



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Gº en Ingeniería en Informática



**Trabajo final del Grado en Ingeniería
Informática:**

**Aplicabilidad de Gazebo en la simulación de
vehículos autónomos**



Presentado por Carmelo Basconcillos Reyero
en julio de 2017
Tutor D. Jesús Enrique Sierra García

Índice de contenido

Índice de ilustraciones.....	2	2.Estructura de directorios.....	30
Índice de tablas.....	3	3.Manual del programador.....	31
I -Planificación.....	4	3.1.Importar máquina virtual.....	32
1.Introducción.....	4	3.2.Instalación de herramientas.....	34
2.Planificación temporal.....	4	3.2.A.Servidor.....	34
2.1.Sprint 1 (6/2/2017 – 19/2/2017).....	4	3.2.B.Moodle.....	36
2.2.Sprint 2 (20/2/2017 - 26/2/2017).....	5	3.3.Versiones de fallos.....	38
2.3.Sprint 3 (27/2/2017 - 12/3/2017).....	5	4.Importación y exportación del curso.....	39
2.4.Sprint 4 (13/3/2017 - 19/3/2017).....	6	4.1.Importación.....	39
2.5.Sprint 5 (20/3/2017 - 26/3/2017).....	6	4.2.Exportación.....	39
2.6.Sprint 6 (27/3/2017 - 2/4/2017).....	7	V -Documentación de usuario.....	40
2.7.Sprint 7 (3/4/2017 - 9/4/2017).....	8	1.Introducción.....	40
2.8.Sprint 8 (17/4/2017 - 23/4/2017).....	8	2.Requisitos de usuarios.....	40
2.9.Sprint 9 (24/4/2017 - 30/4/2017).....	9	2.1.Requisitos software.....	40
2.10.Sprint 10 (1/5/2017 - 7/5/2017).....	10	2.2.Requisitos hardware.....	40
2.11.Sprint 11 (8/5/2017 - 14/5/2017).....	10	3.Manual de usuario.....	40
2.12.Sprint 12 (15/5/2017 - 21/5/2017).....	11	VI -Informe.....	41
2.13.Sprint 13 (22/5/2017 - 28/5/2017).....	12	1.Introducción.....	41
2.14.Sprint 14 (1/6/2017 - 30/6/2017).....	12	1.1.Objetivos.....	42
3.Estudio de viabilidad.....	14	1.2.Metodología.....	42
3.1.Costes personal.....	14	1.3.Contenidos.....	43
3.2.Costes de material.....	14	1.4.Temporalidad.....	43
Costes.....	15	2.Estudio Mercado.....	43
Importe.....	15	2.1.Empresas del sector.....	43
3.3.Viabilidad legal.....	15	2.1.A.JBT.....	43
II -Especificación de requisitos.....	16	2.1.B. Egemin.....	44
1.Introducción.....	16	2.1.C.Electric-80.....	44
2.Objetivos generales.....	16	2.2.Otros.....	45
3.Catalogo de requisitos.....	16	3.Recursos humanos.....	46
4.Especificación de requisitos.....	18	3.1.Conocimientos previos necesarios.....	46
4.1.Diagrama de casos de uso.....	18	4.Plan financiero.....	46
III -Especificación de diseño.....	27	4.1.Costes económicos de licencias y hardware.....	46
1.Introducción.....	27	4.2.Costes económicos de trabajadores.....	47
2.Diseño de datos.....	27	4.3.Previsión de ingresos y amortización.....	47
3.Diseño procedimental.....	28	4.4.Previsión de nuevos gastos.....	47
4.Diseño arquitectónico.....	28	5.Conclusión.....	47
IV -Documentación técnica de programación.....	30	VII -referencias.....	49
1.Introducción.....	30	Bibliografía.....	49





Índice de ilustraciones

Ilustración 1: A.1 Burndown del Sprint 3.....5	Ilustración 20: Nombre de administrador y contraseña.....31
Ilustración 2: A.2 Burndown del Sprint 4.....6	Ilustración 21: Todos los archivos de nuestra máquina virtual.....32
Ilustración 3: A.3 Burndown del Sprint 5.....7	Ilustración 22: VMware Player33
Ilustración 4: A.4 Burndown del Sprint 6.....7	Ilustración 23: Importar máquina virtual.....34
Ilustración 5: A.5 Burndown del Sprint 7.....8	Ilustración 24: Instalación xampp.....35
Ilustración 6: A.6 Burndown del Sprint 8.....9	Ilustración 25: Instalación xampp componentes.....35
Ilustración 7: A.7 Burndown del Sprint 9.....9	Ilustración 26: Instalación xampp carpeta.....36
Ilustración 8: A.8 Burndown del Sprint 10...10	Ilustración 27: Instalación moodle selección idioma.....37
Ilustración 9: A.9 Burndown del Sprint 11...11	Ilustración 28: Instalación moodle ajustes base de datos.....37
Ilustración 10: A.10 Burndown del Sprint 1211	Ilustración 29: Instalación de moodle datos admin.....38
Ilustración 11: A.11 Burndown del Sprint 1312	Ilustración 30: Nombre y contraseña de un usuario.....40
Ilustración 12: Burndown del Sprint 14.....13	Ilustración 31: Visitantes de ROS por países 41
Ilustración 13: A.12 Horas invertidas por semanas.....13	Ilustración 32: Número de robots creados por año.....42
Ilustración 14: B.1 Diagrama de casos de uso del curso.....18	Ilustración 33: Simulador de JBT.....44
Ilustración 15: Diagrama de casos de uso del informe.....19	Ilustración 34: Simulador de Electric-80.....45
Ilustración 16: Diagrama de clases.....27	Ilustración 35: Erle robotics.....45
Ilustración 17: Diagrama de secuencias de nuestro código.....28	Ilustración 36: Puntos en los que se crean robots.....46
Ilustración 18: Estructura curso29	
Ilustración 19: D.1 Ejemplo paquete Catkin. 31	

Índice de tablas

Tabla 1: Tabla de costes.....	15	Tabla 12: Borrar curso.....	23
Tabla 2: Realizar curso.....	19	Tabla 13: Responder al foro.....	23
Tabla 3: Subir dudas al foro.....	20	Tabla 14: Administrar usuarios.....	24
Tabla 4: Realizar encuesta.....	20	Tabla 15: Aceptar usuarios.....	24
Tabla 5: Gestión de usuarios.....	21	Tabla 16: Denegar usuarios.....	24
Tabla 6: Recuperar contraseña.....	21	Tabla 17: Visualizar resultados de la encuesta	
Tabla 7: Login.....	21	25
Tabla 8: Registro.....	22	Tabla 18: Mejorar material didáctico.....	25
Tabla 9: Administrar contenido de moodle...	22	Tabla 19: Crear informe.....	26
Tabla 10: Subir curso.....	22	Tabla 20: Visualizar informe.....	26
Tabla 11: Modificar curso.....	23	Tabla 21: Tabla de fallos.....	38





I - PLANIFICACIÓN

1. Introducción

Vamos a ver como ha ido evolucionando el proyecto durante su realización, así como un estudio de viabilidad donde queremos ver si podemos llevarlo a cabo en el futuro. Para la realización de este proyecto hemos seguido la metodología Scrum, que consiste en tener una estrategia de desarrollo incremental del producto y semanalmente juntar varias fases en paralelo del proyecto. Nos hemos decidido por una metodología Scrum porque así en cada iteración vamos dando valor añadido al proyecto, siempre tenemos versiones nuevas y nuestro trabajo va teniendo más valor. Además de ir marcándonos objetivos semanalmente. La metodología semanal ha sido la siguiente:

- Abrimos Git-Hub
- Creamos una Milestone con duración de una semana.
- Vamos creando issues correspondientes a las tareas, de tal manera que las asociamos a un milestone, un tiempo estimado, un autor, y un proyecto.
- Respecto a la gestión temporal se realiza desde ZenHub, que es una herramienta incluida en el navegador e integrada en GitHub, las tareas se pasan de nuevas a abiertas, esto también nos facilitara gráficas sobre nuestro trabajo.
- Según vamos finalizando las tareas las vamos llevando a tareas cerradas, aquí nos cierra el issue, y podremos ver nuestro gráfico que nos indica el progreso perfecto frente al progreso real.

2. Planificación temporal

El proyecto se ha llevado a cabo marcándonos objetivos semanales, intentando cumplirlos. Como veremos a continuación, alguna semana en la que disponíamos de más tiempo hemos incluido alguna tarea extra y por el contrario en otra hemos dejado una determinada tarea retrasada. Veremos el trabajo realizado cada semana.

2.1. Sprint 1 (6/2/2017 – 19/2/2017)

Es una toma de contacto con la gran cantidad de herramientas con las que vamos a trabajar. Lo primero y fundamental es la creación de una máquina virtual Ubuntu 16.04. Sobre ella se va a instalar el software de Gazebo, se van a seguir distintos tutoriales que hemos encontrado por Internet, con el inconveniente de que el primero lleva una versión de ubuntu más antigua, y ralentiza mucho la máquina virtual. Tras analizar algunas herramientas de gestión de proyectos como Git, Bitbucket o GitHub, se decide trabajar con GitHub como gestor de versiones y con ZenHub como gestor de proyectos

Tras ello vamos a realizar diferentes tutoriales de la página oficial de gazebo tanto de su manejo como del modelado de robots.

Cumplido: No sabemos si hemos cumplido el objetivo, dado que no sabíamos como usar el repositorio, y tuvimos que cambiar los milestones para que nos generasen gráficas. Aún con esto el imprevisto de reinstalar la máquina virtual nos habría retrasado casi con seguridad.

2.2. Sprint 2 (20/2/2017 - 26/2/2017)

Esta semana nos hemos marcado como objetivo poder realizar nuestro primer robot en gazebo. Para ello hemos hecho un primer diseño en papel de lo que queríamos (aconsejados por el tutor), hemos realizado diferentes tutoriales de SDF y hemos llevado a cabo la realización y puesta en movimiento del mismo.

Cumplido: Esta semana hemos dado un gran paso, puesto que hemos entendido como funcionan los robots, donde y como crear las librerías para que nuestros robots aparezcan en el programa, y hemos sabido manejar bien gazebo.

2.3. Sprint 3 (27/2/2017 - 12/3/2017)

Una vez ya manejado el entorno de gazebo, durante esta semana nos hemos dedicado a informarnos y documentarnos sobre el sistema operativo ROS; después de un fallo con la versión de ubuntu hemos conseguido instalar ROS con todas las librerías kinetic, también hemos visto el uso del manejador de fuerzas Rviz, y hemos realizado los primeros tutoriales de ROS, donde hemos aprendido como funciona ROS, sistema publicador/suscriptor.

Cumplido: Hemos cumplido los objetivos, aunque la realización de los tutoriales nos ha llevado más tiempo de lo esperado, puesto que en ocasiones nos dan errores, no nos compila parte, etc. Primera semana que nos deja rescatar la gráfica de trabajo, y como la realización de tutoriales era una carga muy pesada se ve como las tareas se realizan muy al final.



Ilustración 1: A.1 Burndown del Sprint 3





2.4. Sprint 4 (13/3/2017 - 19/3/2017)

Esta semana es la de los primeros pasos reales en ROS, hemos conseguido lanzar un mundo con un robot que en el momento que se lanza es capaz de escribirnos un “Hola mundo”. Es la primera piedra sobre la que se sostiene ROS, el publisher, donde publica los mensajes tu robot.

Cumplido: Hemos cumplido los objetivos, y además los tiempos marcados son muy similares a la línea de la gráfica perfecta, por lo que esta semana acabamos muy contentos con nuestro trabajo. Hemos usado eclipse para que sea nuestra IDE.

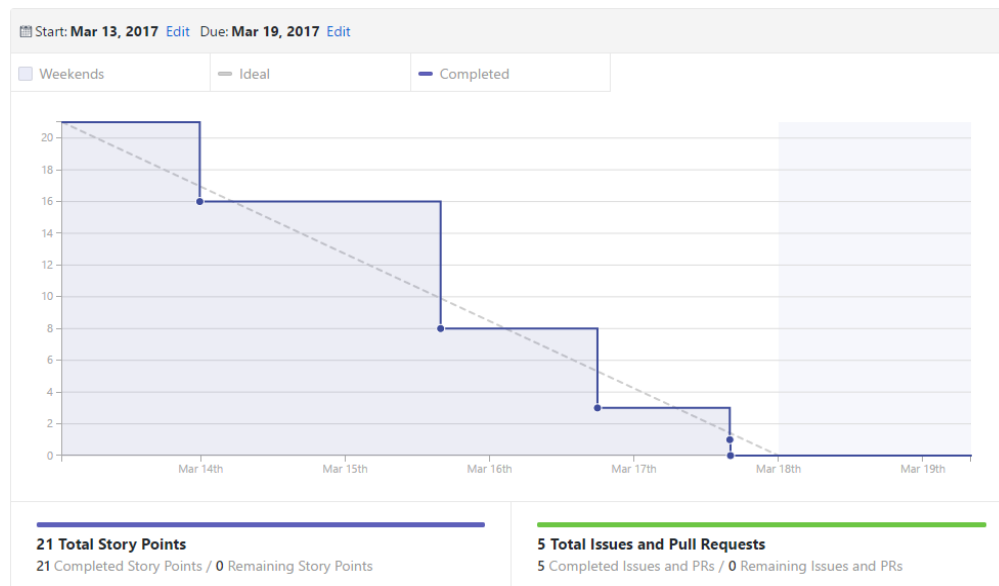


Ilustración 2: A.2 Burndown del Sprint 4

2.5. Sprint 5 (20/3/2017 - 26/3/2017)

Durante esta semana hemos cargado menos horas de trabajo, el objetivo era que nuestro robot recibiese y mandase mensajes por medio de ROS, un objetivo irreal es que interpretase ordenes, y dimos los primeros pasos para ello.

Cumplido: Hemos cumplido los objetivos, aunque no todos, puesto que el robot no interpreta ordenes todavía, damos los primeros pasos y lo dejamos para siguientes sprints. La curva de trabajo vuelve a ser casi idónea.



Ilustración 3: A.3 Burndown del Sprint 5

2.6. Sprint 6 (27/3/2017 - 2/4/2017)

Esta semana pasada ya hemos conseguido que nuestro robot escuche y hable, vamos a probar con más robots, en este caso con un jackal y con un turtlebot, para más tarde insertar estos comandos a nuestros robots. Realizaremos tutoriales para ver si aprendemos como hacer que el robot interprete ordenes, de momento sin éxito.

Cumplido: Hemos realizado los tutoriales, y hemos comprendido mejor como funcionan, pero seguimos sin ver como hacer que un robot lanzado por nosotros interprete ordenes.

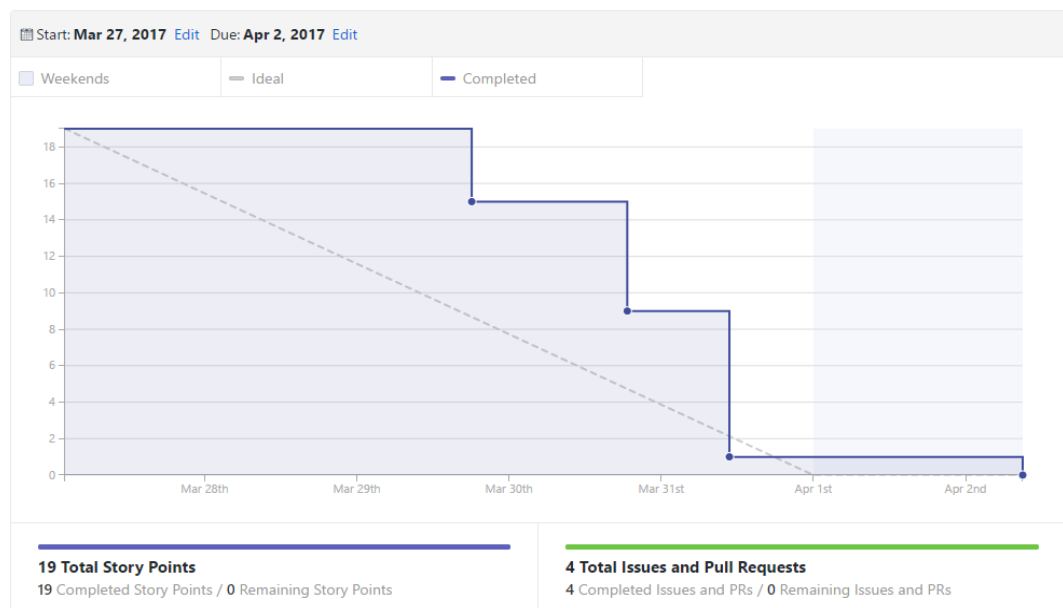


Ilustración 4: A.4 Burndown del Sprint 6





2.7. Sprint 7 (3/4/2017 - 9/4/2017)

En esta semana el objetivo real era que nuestro robot, uno generado por nosotros interpretase movimiento y conseguir moverlo. Después de perder mucho tiempo no hemos conseguido que nuestro robot se mueva por medio de ROS, debe de haber algún fallo en el modelado del robot que no hemos sabido ver.

Cumplido: Esta semana no hemos cumplido el objetivo, hemos tenido fallos, y no nos ha funcionado lo que queríamos, además por culpa de esto nuestra gráfica de trabajo ha quedado retrasada porque no hemos cumplido los plazos.

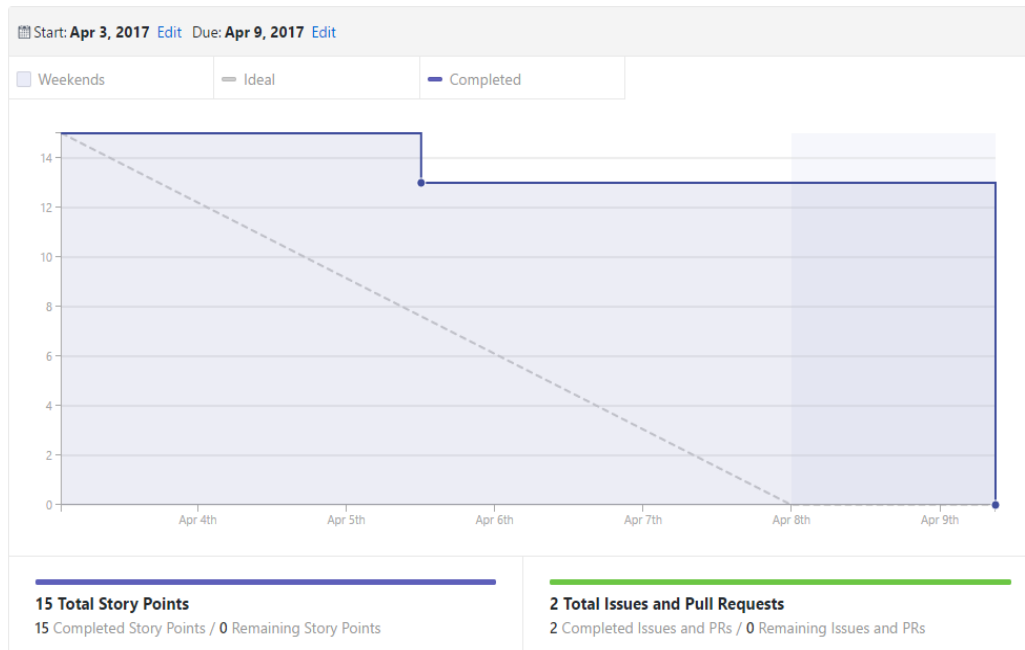


Ilustración 5: A.5 Burndown del Sprint 7

2.8. Sprint 8 (17/4/2017 - 23/4/2017)

En esta semana queremos lograr el objetivo primordial del proyecto, que nuestro robot consiga interpretar ordenes, en este caso de movimiento, tanto para sus ruedas como para sus brazos articulados.

Cumplido: Esta semana conseguimos nuestro objetivo, si bien los plazos de tiempo como se observa en la gráfica son tardíos, puesto que finalmente dedicamos más tiempo a ello, logramos conseguir el movimiento en nuestro robot por medio de ROS.

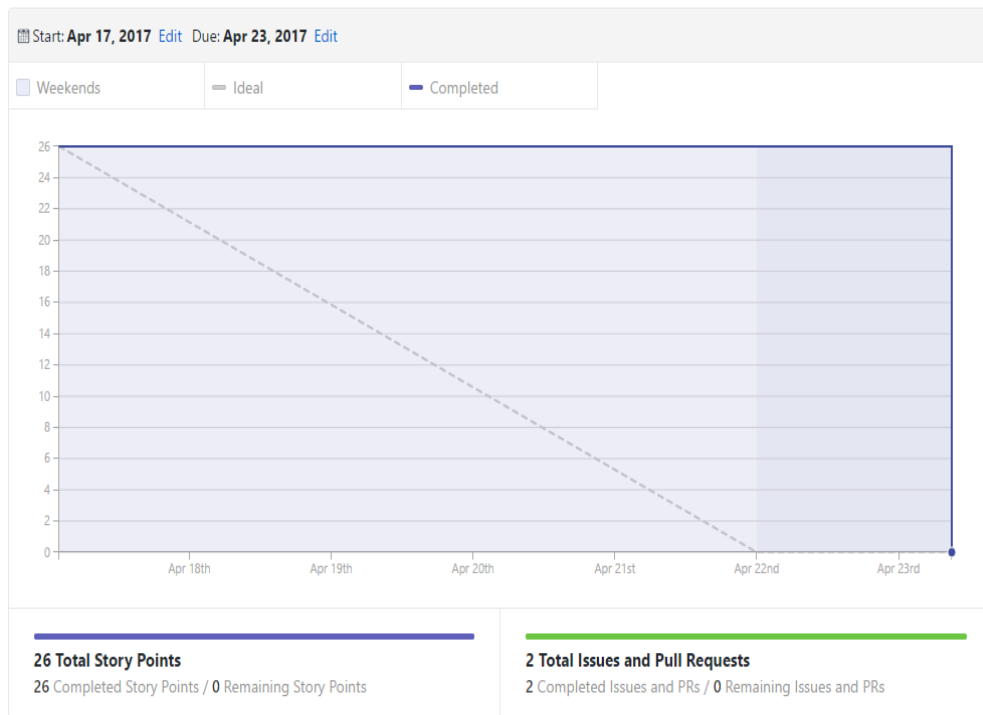


Ilustración 6: A.6 Burndown del Sprint 8

2.9. Sprint 9 (24/4/2017 - 30/4/2017)

Empezaremos la realización del manual de usuario que más tarde se entregará como anexo a la memoria, uno de los objetivos del proyecto para reducir la costosa curva de aprendizaje de ROS.

Cumplido: Manual de usuario completado de manera eficiente, con esto se cumpliría uno de los objetivos del proyecto. La caída de la curva es muy pronunciada en el gráfico porque solo teníamos una tarea grande, no vale de mucho en este caso el gráfico.

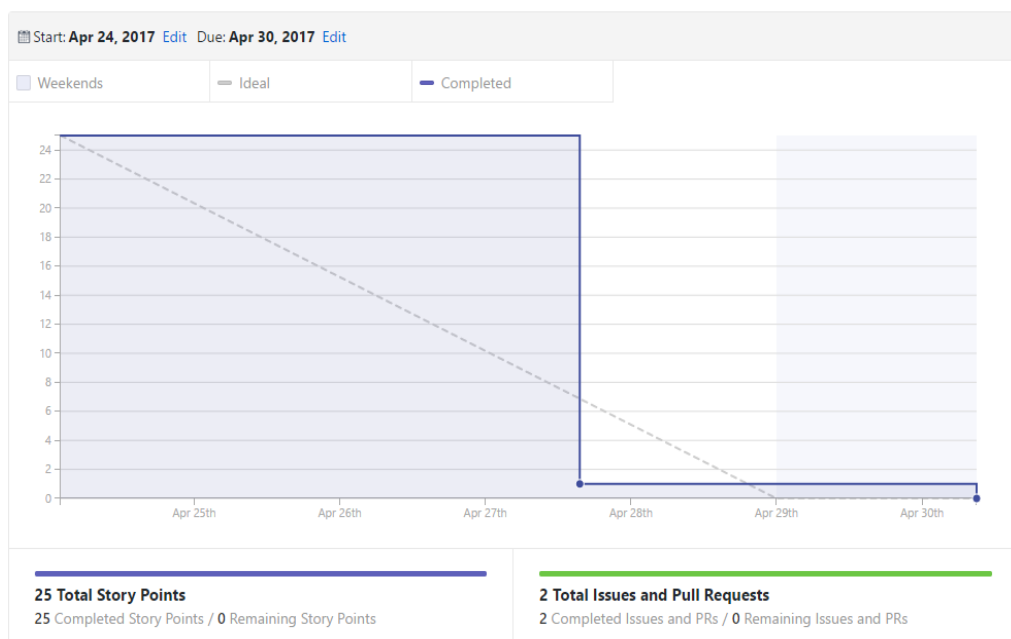


Ilustración 7: A.7 Burndown del Sprint 9





2.10. Sprint 10 (1/5/2017 - 7/5/2017)

Semana que dedicaremos expresamente a hacer memoria, para que nos quede una gráfica más real hemos cogido todos los puntos de la memoria que hemos ido realizando y los hemos hecho issues, así podremos llevar un mejor seguimiento del tiempo.

Cumplido: Nos propusimos en dos semanas de trabajo completar la memoria, y hemos llegado a hacer la mitad más o menos, por lo cual el objetivo está cumplido.



Ilustración 8: A.8 Burndown del Sprint 10

2.11. Sprint 11 (8/5/2017 - 14/5/2017)

En esta semana queremos empezar lo que sería nuestro informe a entregar finalmente, después de hablar con nuestro Tutor hemos acordado los puntos, y llevar a cabo los primeros pasos para la realización del mismo.

Cumplido: La reunión fue satisfactoria, salió todo adelante y tenemos la primera parte del entregable finalizada.

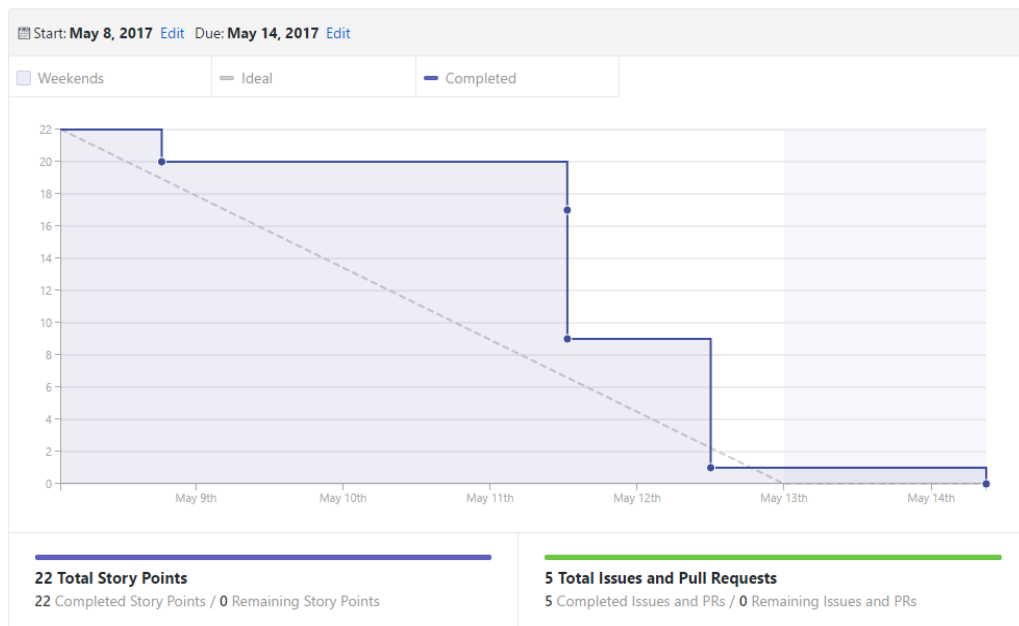


Ilustración 9: A.9 Burndown del Sprint 11

2.12. Sprint 12 (15/5/2017 - 21/5/2017)

En esta semana queremos terminar lo que sería nuestro informe a entregar finalmente, después de empezarlo la semana pasada queremos llegar a completarlo, o casi completarlo esta semana.

Cumplido: No se ha llegado a completar el informe, pero hemos realizado puntos muy importantes a seguir en él, trabajo constante en la semana.

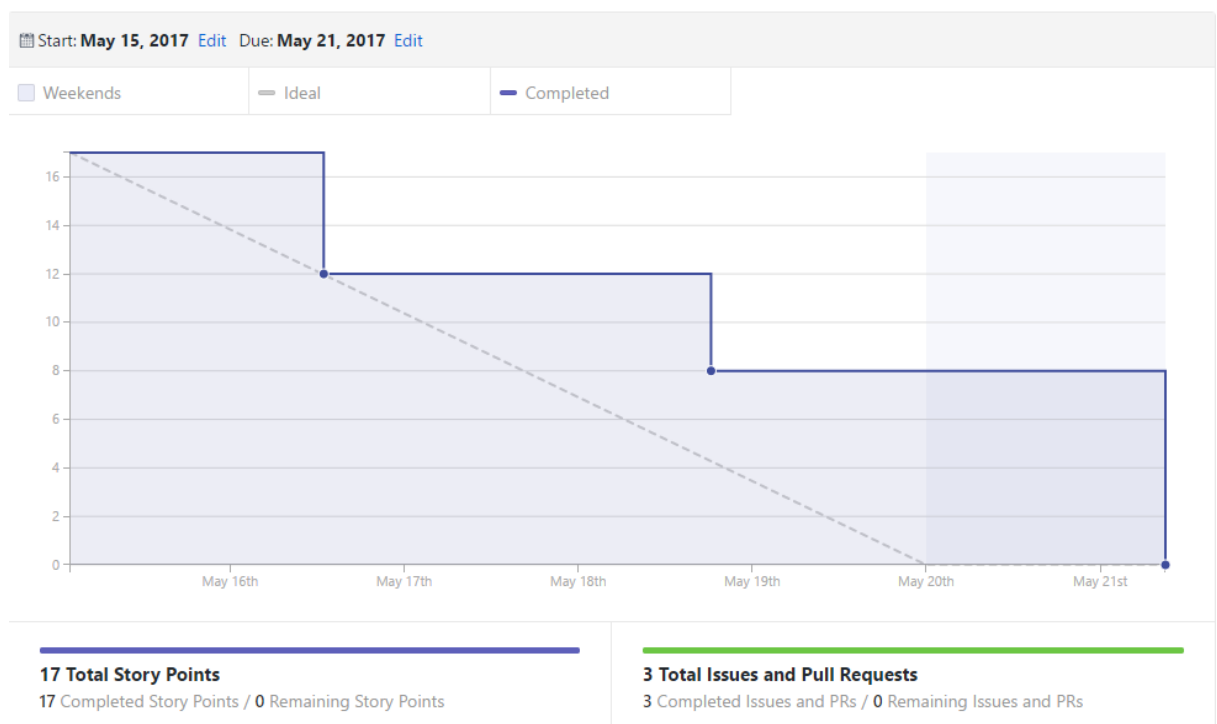


Ilustración 10: A.10 Burndown del Sprint 12



2.13. Sprint 13 (22/5/2017 - 28/5/2017)

En esta semana queremos dejar todo hecho en cuanto a la memoria para centrarnos en los anexos la última semana de trabajo.

Cumplido: Hemos conseguido acabar la memoria, y ya solo quedan los anexos para la próxima semana, el trabajo ha sido bueno y constante, según muestra el grafo.



Ilustración 11: A.11 Burndown del Sprint 13

2.14. Sprint 14 (1/6/2017 - 30/6/2017)

En estas 4 semanas tiene que estar acabado todo el proyecto y listo para entregar

Cumplido: Hemos conseguido acabar el proyecto, y el trabajo ha sido un éxito.

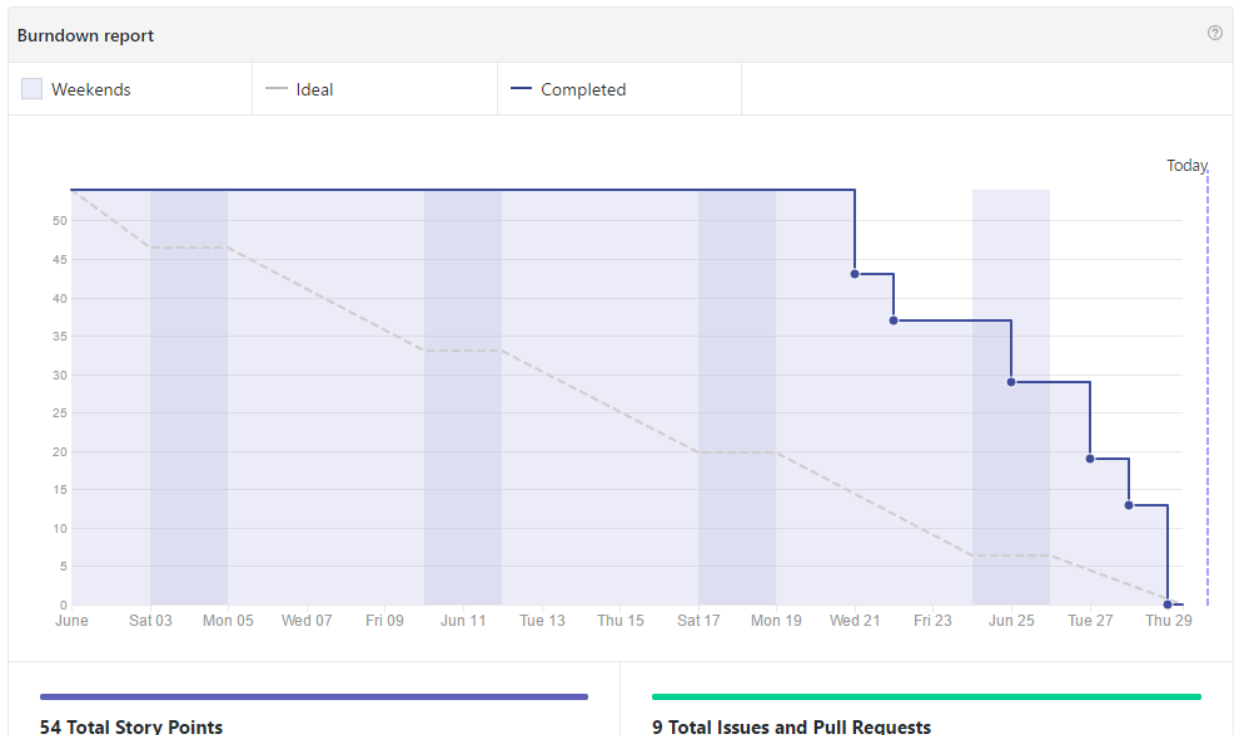


Ilustración 12: Burndown del Sprint 14

Según se puede apreciar en la siguiente gráfica, las últimas 8 semanas de trabajo (de las primeras no guardé información) se ve un trabajo semanal constante, donde la media es de 22 horas por semana, lo que son sin contar los fines de semana unas 4,5 horas al día con el proyecto. El último sprint es de 4 semanas por eso tiene un pico en la gráfica más elevado.

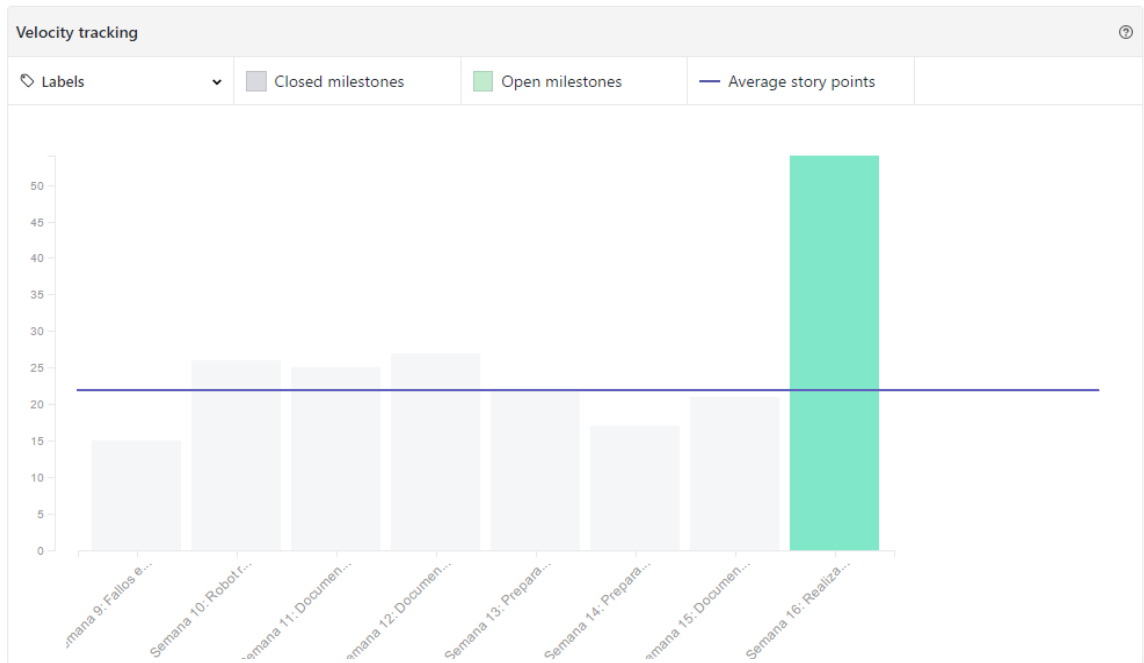


Ilustración 13: A.12 Horas invertidas por semanas





3. Estudio de viabilidad

En esta sección vamos a analizar si se puede apostar o no por este proyecto en un futuro, vamos a tener en cuenta diferentes puntos de vista, como la viabilidad económica y la legal. Sin duda ambos factores son determinantes para seguir con el proyecto o al menos para que podamos lucrarnos de ello.

3.1. Costes personal

Para un único trabajador durante 4 meses y medio, podemos usar las horas invertidas en github para hacer con más exactitud los cálculos:

Es necesario contratar a un trabajador con experiencia en C++ y C, que tenga cierta soltura para el aprendizaje de nuevos lenguajes (como SDF), estimamos que el sueldo del trabajador será de 12€ /hora netos, por lo que el 32,1% de cotización a la Seguridad Social le corresponde a la empresa.

Teniendo en cuenta que se han usado 415 horas* 12€/hora = 4980€ a pagar al trabajador.

A esta cuanta debemos añadirle el valor que se deberá llevar la Seguridad Social que equivaldrá a:

23,6 %: Por Seguridad Social.

5,5 %: Por desempleo.

0,6 %: Por formación profesional.

0,2 %: Por Fogasa.

Total: 29,9%

Y esto equivale a:

$$4980€ * 0,299 = 1489€$$

3.2. Costes de material

Por software solo tenemos gasto en el editor de video Moravi PTT converter, cuyo coste de licencia ilimitada es de 22.95€, y el Office 365, que tiene un coste de 7€ al mes*5 meses= 35€, el resto de software no tiene coste ya que todos lo demás que usamos no necesita licencias de pago, en el caso de VMware su coste de licencia es de 251,95€, aunque usamos una versión no comercial para el proyecto. Por el hardware yo he usado mi propio equipo, cuyas características son Intel Core i7 8 Gb de RAM y un disco duro de 1TB, que tiene un valor aproximado de 900€, se considera que la amortización de un equipo ronda los 5 años y nosotros lo hemos usado 4 meses y medio por lo tanto el coste en hardware ha sido:

$$900€ / (12\text{Meses} * 5 \text{ años}) = 15€/mes$$

$$4,5\text{MesesDeUso} * 15€/mes = 60€$$

Los gastos mensuales de Internet son de 35€ mensuales, suponiendo 140€ al finalizar el proyecto.

Costes	Importe
Personal	4980€
Seguridad Social	1489€
Material	60€
Internet	140€
Licencias	309,9€
Total:	6978,9€

Tabla 1: Tabla de costes

3.3. Viabilidad legal

En este punto lo que vamos a hacer es analizar las librerías que hemos usado en nuestro proyecto. Buscar los tipos de licencias que tienen cada uno, luego analizar, dependiendo de las compatibilidades, que tipos de licencia podemos aplicar nuestro proyecto, las licencias de los programas utilizados tienen las siguientes licencias:

ROS: BSD.

SDF: Apache 2.0.

Vmware: Licencia de Pago. Una alternativa es Virtual Box de licencia GPL.

Ubuntu: GPL y otras licencias libres.

Gazebo: Licencia Apache 2.0.

Moodle: Licencia GNU- GPL, licencia gratuita.

Microsoft PowerPoint: Licencia de pago. Una alternativa es Impress de licencia LGPL

v3.





II - ESPECIFICACIÓN DE REQUISITOS

1. Introducción

El objetivo de este apartado es la definición de las características que se desea que la aplicación tenga al final de su desarrollo. Para ello fijaremos los objetivos que se intentan cumplir y, más formalmente, se definirán los requisitos funcionales y no funcionales de la aplicación. Finalmente se profundizará en los casos de uso derivados de la funcionalidad que se desea.

2. Objetivos generales

Tal y como se trató en la memoria, el proyecto cuenta con dos objetivos claramente definidos.

Principalmente el objetivo es aprender a usar esta tecnología, para más adelante conseguir realizar un informe donde se saquen unas buenas conclusiones, y se consiga diseñar un curso online con el cual la implantación de esta tecnología sea mucho más sencilla.

3. Catalogo de requisitos

RF-1: Modelar nuestro robot dentro de la plataforma Gazebo

RF-1: Obtener un primer diseño del robot

RF-2: Identificar sus partes

RF-3: Modelar el diseño a ordenador poniendo los tipos de objeto en función a su utilidad

RF-4: Probar el funcionamiento del robot en la plataforma Gazebo para su correcto funcionamiento

RF-2: Conectarnos con el robot creado por medio de ROS

RF-1: Crear un mundo con sus elementos esenciales

RF-2: Lanzar nuestro robot dentro del mundo

RF-3: Lanzar nodos suscriptor y publicador

RF-4: Realizar programación para movimiento del robot

RF-5: Almacenar ordenes en scripts con movimientos

RF3: Diseñar un informe de viabilidad para una empresa

RF-1: Redactar los puntos importantes del informe

RF-2: Documentar todos los puntos en función a los conocimientos adquiridos

RF-3: Visualizar informe

RF-4: Diseñar un curso online

RF-1: Realizar página moddle

RF-2: Realizar powerpoints para la presentación

RF-3: Realizar videotutoriales

RF-4: Incluir temario en la página

RF-5 Gestionar página

RF-5: Realizar curso

RF-1: Visualizar cursos

RF-2 Realizar curso

RF-3: Realizar encuesta

RF-4 Subir dudas al foro





4. Especificación de requisitos

4.1. Diagrama de casos de uso

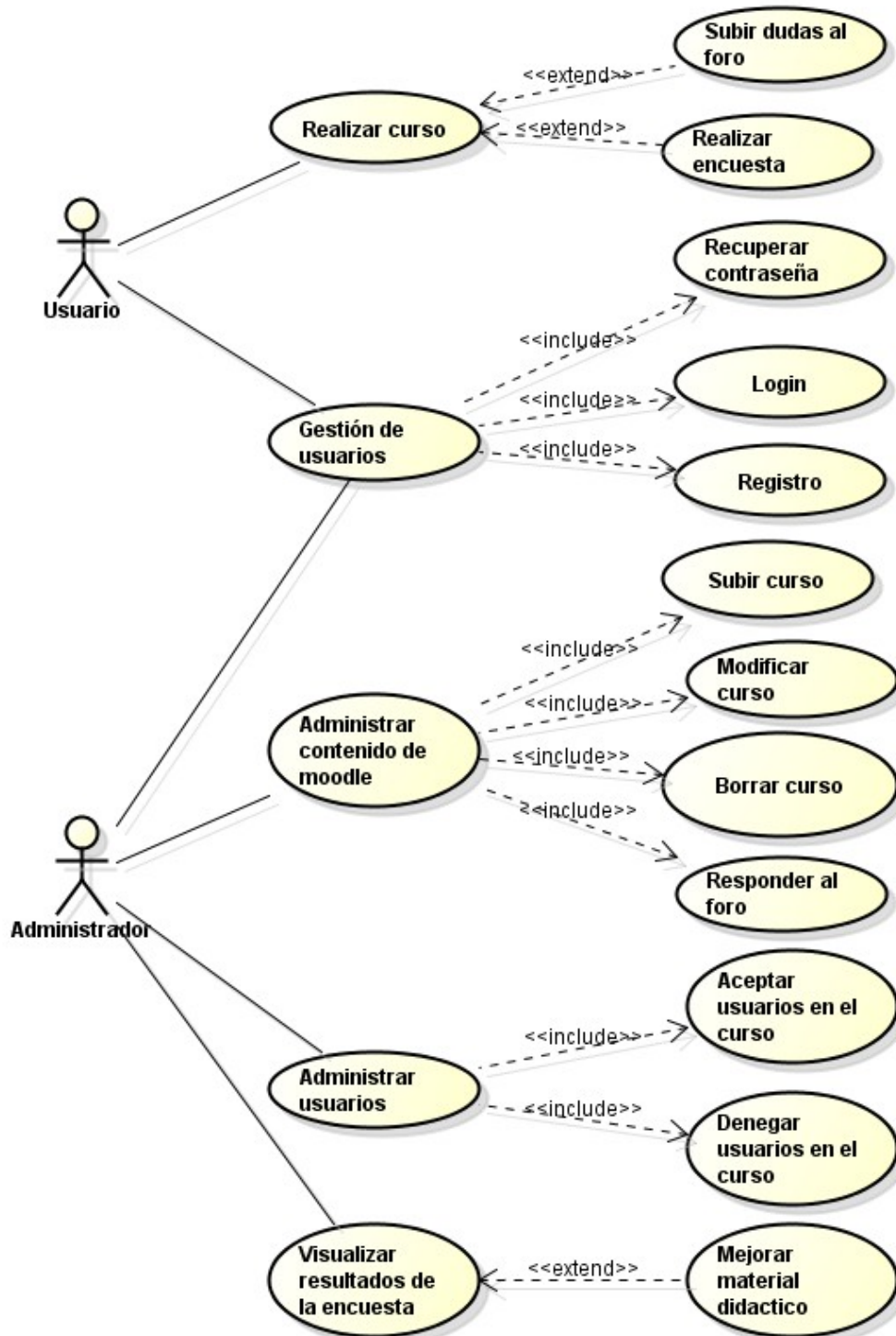


Ilustración 14: Diagrama de casos de uso del curso

En la ilustración 14, podemos observar el diagrama de casos de uso del curso, en la ilustración 15 vemos el diagrama del informe, a continuación vamos a ver más detalladamente cada caso de uso. Son dos los actores que podemos encontrar en el diagrama de casos de uso, el administrador se ocupa de realizar todas las acciones, y el usuario únicamente puede registrarse y hacer acciones referentes a realizar el curso.

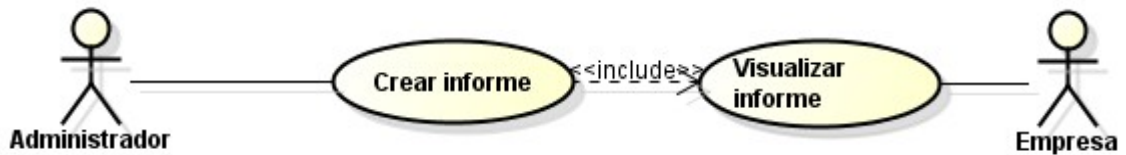


Ilustración 15: Diagrama de casos de uso del informe

Caso de uso	Realizar curso
Requisitos	RF-5 RF-5.1 RF-5.2 RF-5.3 RF-5.4
Descripción	Debemos realizar un curso online
Precondiciones	El curso debe de estar creado y nosotros registrados en la aplicación
Acciones	1. El usuario se registra en un curso 2. El usuario realiza el curso
Postcondiciones	El usuario deberá de valorar el curso
Excepciones	El administración puede bloquear usuarios
Importancia	Alta

Tabla 2: Realizar curso





Caso de uso	Subir dudas al foro
Requisitos	RF-5.1 RF-5.2 RF-5.4
Descripción	El usuario sube al foro las dudas que hay tenido durante el curso
Precondiciones	El foro debe de estar creado y abierto para la edición
Acciones	1. Visualizar curso 2. Realizar curso 3. Subir duda al foro
Postcondiciones	El administrador debe responder esta duda
Excepciones	Para poder realizar una entrada en el foro debes de estar registrado
Importancia	Media

Tabla 3: Subir dudas al foro

Caso de uso	Realizar encuesta
Requisitos	RF-5.1 RF-5.2 RF-5.3
Descripción	El usuario realiza una encuesta sobre el curso
Precondiciones	El administrador tiene que haber creado la plataforma y habilitado la encuesta
Acciones	1. Crear cursos y meter el material 2. Gestionar los cursos 3. Visualizar cursos 4. Realizar el curso 5. Valorar el curso
Postcondiciones	Ninguna
Excepciones	Debe de estar registrado
Importancia	Alta

Tabla 4: Realizar encuesta

Caso de uso	Gestión de usuarios
Requisitos	RF-4.5
Descripción	El administrador gestiona los permisos de los usuarios
Precondiciones	Debe de haber un usuario interesado en el curso
Acciones	1. Buscar el usuario en moodle 2. Gestionar sus permisos
Postcondiciones	El administrador puede dar unos valores y luego modificarlos
Excepciones	Ninguna
Importancia	Baja

Tabla 5: Gestión de usuarios

Caso de uso	Recuperar contraseña
Requisitos	RF-4.5 RF-5.1
Descripción	Si perdemos la contraseña moodle nos ayuda a recuperarla con el correo
Precondiciones	Haber perdido la contraseña
Acciones	1. Dan en recuperar contraseña 2. Introducir tu correo 3. Reiniciar contraseña
Postcondiciones	Restaurar la contraseña
Excepciones	Ninguna
Importancia	Baja

Tabla 6: Recuperar contraseña

Caso de uso	Login
Requisitos	RF4.5 RF 5.1
Descripción	Meter tus credenciales en la aplicación
Precondiciones	Estar registrado en moodle
Acciones	1. Introducir nombre usuario 2. Introducir contraseña
Postcondiciones	Ninguna
Excepciones	Ninguna
Importancia	Media

Tabla 7: Login



Caso de uso	Registro
Requisitos	RF4.5 RF 5.1
Descripción	Nos registramos en la aplicación de moodle para poder acceder a sus contenidos
Precondiciones	Ninguna
Acciones	1. Introducir datos personales 2. Elegir nombre usuario 3. Elegir contraseña
Postcondiciones	Deberemos loguearnos después con nuestras credenciales
Excepciones	No puedes registrar dos emails
Importancia	Media

Tabla 8: Registro

Caso de uso	Administrar contenido de moodle
Requisitos	RF4 RF 4.1
Descripción	Administrar el curso de moodle
Precondiciones	Tener el curso creado
Acciones	1. Administrar diferentes funciones del curso, actualizar contenidos etc
Postcondiciones	Ninguna
Excepciones	Ninguna
Importancia	Alta

Tabla 9: Administrar contenido de moodle

Caso de uso	Subir curso
Requisitos	RF4 RF 4.1 RF4.2 RF 4.4
Descripción	Creamos un curso y lo subimos a la plataforma
Precondiciones	El administrador debe haber realizado el material didáctico correspondiente
Acciones	1. Crear curso 2. Elegir la manera de llevar el curso(temas, semanas...) 3. Subir material al curso
Postcondiciones	Ninguna
Excepciones	No permite la subida de vídeos grandes, usaremos youtube
Importancia	Alta

Tabla 10: Subir curso

Caso de uso	Modificar curso
Requisitos	RF4 RF 4.1 RF4.2 RF 4.4
Descripción	Modificamos un curso y lo actualizamos en la plataforma
Precondiciones	Tener creado un curso
Acciones	1. Elegir material a modificar 2. Realizamos nuevo material didáctico 3. Actualizamos en la plataforma
Postcondiciones	Ninguna
Excepciones	No permite la subida de vídeos grandes, usaremos youtube
Importancia	Media

Tabla 11: Modificar curso

Caso de uso	Borrar curso
Requisitos	RF4.5 RF 4
Descripción	Borramos un tema o el curso entero de moodle
Precondiciones	Tener creado un curso
Acciones	1. Elegir material a borrar 32. Borramos en la plataforma
Postcondiciones	Ninguna
Excepciones	Ninguna
Importancia	Media

Tabla 12: Borrar curso

Caso de uso	Responder al foro
Requisitos	RF4.5
Descripción	El administrador responde a las preguntas de los usuarios en el foro
Precondiciones	Un usuario ha debido de realizar alguna pregunta
Acciones	1. Documentarnos sobre el tema preguntado 2. Responder a la pregunta
Postcondiciones	Ninguna
Excepciones	Si no conocemos el tema podremos ayudar al usuario
Importancia	Media

Tabla 13: Responder al foro



Caso de uso	Administrar usuarios
Requisitos	RF4.5
Descripción	El administrador puede administrar a los usuarios dentro de su curso
Precondiciones	Ninguna
Acciones	1. El administrador decide sobre los usuarios
Postcondiciones	Podremos cambiar las decisiones tomadas sobre el usuario
Excepciones	Ninguna
Importancia	Media

Tabla 14: Administrar usuarios

Caso de uso	Aceptar usuarios
Requisitos	RF4.5
Descripción	El administrador acepta al usuario en el curso
Precondiciones	El usuario ha tenido que solicitar unirse al curso
Acciones	1. Aceptar al usuario
Postcondiciones	Podremos cambiar las decisiones tomadas sobre el usuario
Excepciones	Ninguna
Importancia	Media

Tabla 15: Aceptar usuarios

Caso de uso	Denegar usuarios
Requisitos	RF4.5
Descripción	El administrador rechaza al usuario en el curso
Precondiciones	El usuario ha tenido que solicitar unirse al curso
Acciones	1. Rechazar al usuario
Postcondiciones	Podremos cambiar las decisiones tomadas sobre el usuario
Excepciones	Ninguna
Importancia	Media

Tabla 16: Denegar usuarios

Caso de uso	Visualizar resultados de la encuesta
Requisitos	RF4.5 RF 5.3
Descripción	El administrador visualiza los resultados de la encuesta
Precondiciones	Los usuarios han debido de realizar el curso
Acciones	1. El usuario realiza la encuesta 2. El administrador ve los resultados de la encuesta
Postcondiciones	Ninguna
Excepciones	Ninguna
Importancia	Media

Tabla 17: Visualizar resultados de la encuesta

Caso de uso	Registro
Requisitos	RF4 RF 4.2 RF 4.3 RF 4.4 RF 4.5
Descripción	El administrador mejora los contenidos del curso en función al resultado de la encuesta
Precondiciones	Los usuarios deben haber realizado la encuesta
Acciones	1. Las encuestas o el foro refleja fallos o partes no satisfactorias 2. El administrador mejora el material didáctico 3. El administrador sube el material al curso
Postcondiciones	Deberemos loguearnos después con nuestras credenciales
Excepciones	No puedes registrar dos emails
Importancia	Alta

Tabla 18: Mejorar material didáctico



Caso de uso	Crear informe
Requisitos	RF-3 RF-3.1 RF-3.2 RF-3.3
Descripción	Informe de viabilidad de la herramienta Gazebo-ROS en una empresa de logística interna.
Precondiciones	Haber trabajado con la herramienta
Acciones	1. Fijar puntos de interés del informe 2. Documentar los puntos
Postcondiciones	Ninguna
Excepciones	Ninguna
Importancia	Baja

Tabla 19: Crear informe

Caso de uso	Visualizar informe
Requisitos	RF-3.4
Descripción	La empresa visualiza el informe realizado por el administrador
Precondiciones	Ninguna
Acciones	1. Visualizar el informe 2. Tomar medias de implantación en empresa
Postcondiciones	Ninguna
Excepciones	Ninguna
Importancia	Media

Tabla 20: Visualizar informe

III - ESPECIFICACIÓN DE DISEÑO

1. Introducción

En este apartado vamos a explicar que hemos usado para la resolución del problema en cuanto a herramientas de diseño.

Para la elaboración de los diagramas de paquetes y de clases hemos usado UML como lenguaje de modelado y el programa Astah para implementar los diagramas de clases.

2. Diseño de datos

En la ilustración número 16 podemos ver el diagrama de clases de nuestro código, la clase ros es una librería de ros, la cual no hemos creado nosotros, pero si definimos atributos y métodos de esta clase dado que incluimos esta librería en el Listner.

Listner es quien se encarga de la comunicación con ROS.

Comandos es la función intermedia que comunica lo recibido en el subscriptor de ROS y nos lanza el comando a la clase MiRobot que interpreta la orden.

La clase common y physics son dos clases de Gazebo de las cuales hereda MiRobot, en MiRobot se producen las ordenes al robot para que se mueva.

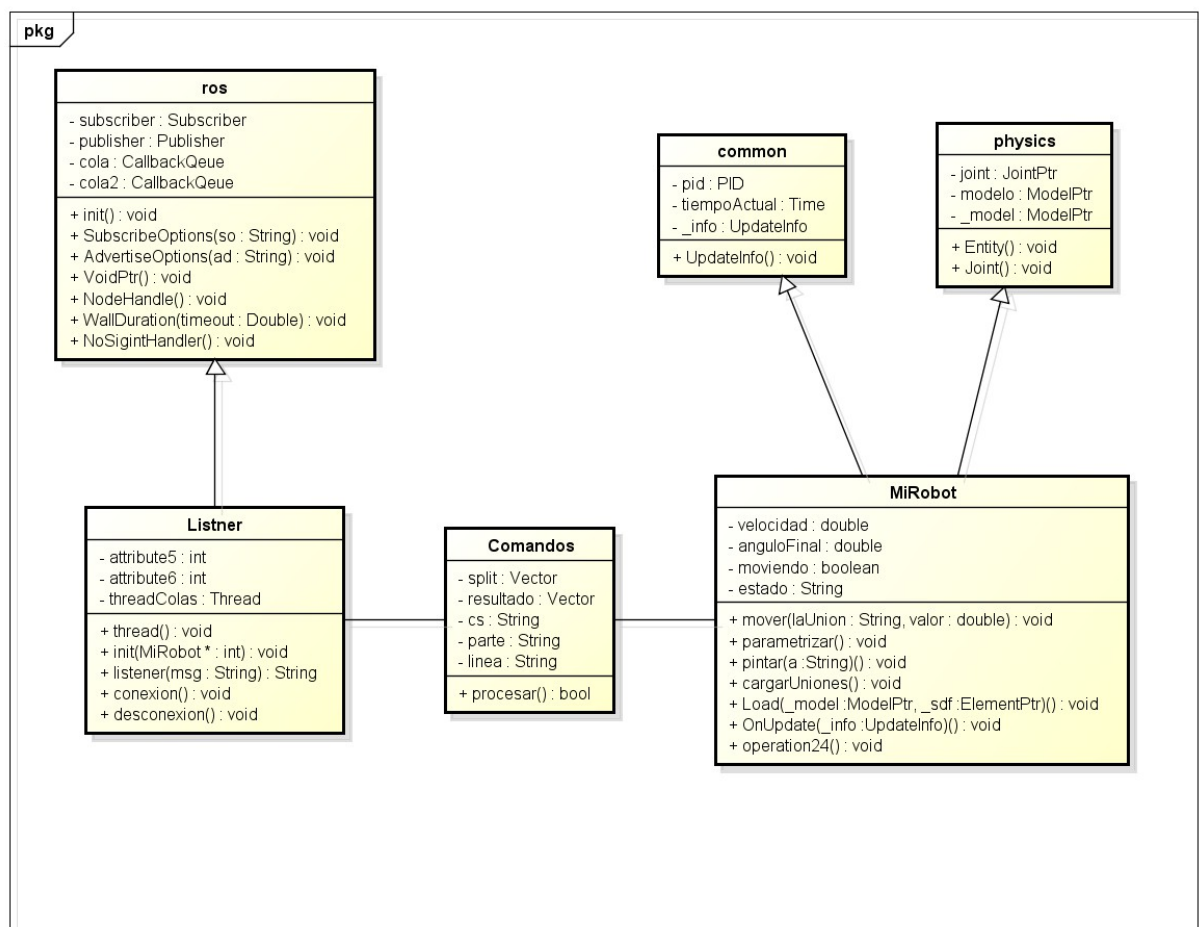


Ilustración 16: Diagrama de clases





3. Diseño procedimental

En cuanto al diseño procedimental lo basaremos en el código de ejemplo que hemos creado para el curso, y para una posible simulación. Se puede ver en la ilustración 17 como se llaman las clases entre si.

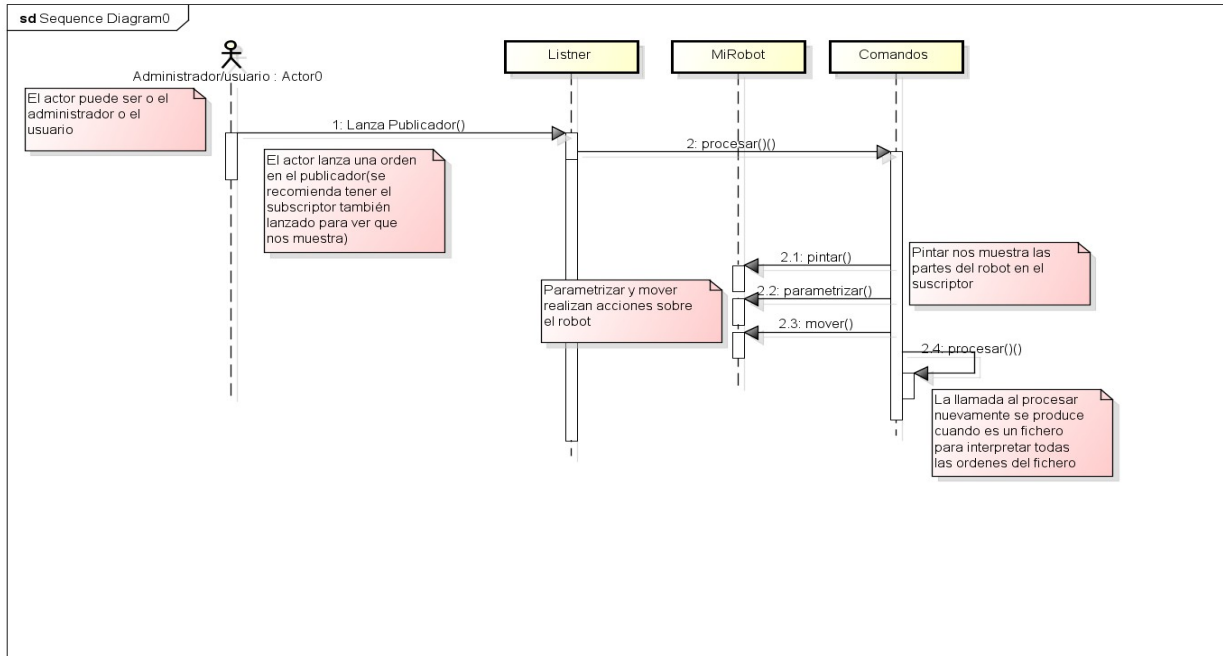


Ilustración 17: Diagrama de secuencias de nuestro código

En cuanto al curso, el diseño que hemos seguido ha sido primero elaborar el material didáctico por medio de powerpoint, hemos definido los puntos que creíamos más importantes y lo hemos redactado por temas. Después hemos hecho con estos powerpoints videotutoriales para que el aprendizaje sea más dinámico. Por último hemos añadido material complementario con ejemplos de uso que funcionan.

En la parte del informe hemos realizado un informe analítico, que tiene como objetivo justificar una decisión ya proyectada. Es una justificación de la viabilidad de esta herramienta en empresas del sector. [1]



4. Diseño arquitectónico

En cuanto al diseño arquitectónico del curso se organiza en secciones o temas, a los que el administrador puede dar títulos. Cada tema consiste en actividades, recursos y etiquetas. Este tipo de diseño está basado en objetivos y cada objetivo puede necesitar diferentes cantidades de tiempo para completar.[2]

Se puede ver que todos los temas cuentan con un videotutorial, un powerpoint, y los que procede cuentan también con material complementario. En la introducción aparecen las APIS más importantes del proyecto que se han de usar durante el curso.

-  Guía docente
-  Avisos
-  Foro general de la asignatura
-  Página del lenguaje SDF
-  Página del sistema operativo ROS
-  Página del simulador Gazebo
-  Información adicional ROS




Tema 1

-  Vídeo tema 1: Conceptos básicos
-  Powerpoint tema 1

Tema 2

-  Vídeo tema2: Instalación de recursos
-  Powerpoint tema 2

Tema 3

-  Vídeo tema 3: Preparación del entorno
-  Powerpoint tema 3
-  mirobot.launcher

Tema 4

-  Vídeo tema 4: Manejo simple de gazebo
-  Powerpoint tema 4
-  mirobot.world

Tema 5

-  Vídeo tema5: Modelaje de nuestro robot
-  Powerpoint tema 5
-  Ejemplo primer AGV
-  Ejemplo robot con sensores

Ilustración 18: Estructura curso





IV - DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

1. *Introducción*

Esta sección esta dirigida a otros desarrolladores, de modo que puedan continuar el proyecto y entenderlo. En el se describen en detalle el funcionamiento del proyecto y los aspectos que podrán mejorarse o modificarse en el futuro.

2. *Estructura de directorios*

Existen tres carpetas principales en el proyecto:

- Carpeta Devel: La carpeta devel es la ubicación por defecto donde los ejecutables y librerías van antes de ser instaladas.
- Carpeta bin: contiene los ejecutables del paquete, los que se podrán lanzar como nodos.
- Carpeta lib: contiene los archivos de librerías (.lib, .so). Aquí se crea la librería de comunicación con el robot cuando compilamos nuestro código.(Se encuentra dentro de Devel.)
- Build: Contiene los archivos de compilación internos del paquete.
- Src: Contendrá nuestros códigos de los programas que creemos (.cpp), que al compilarlos crearán ejecutables en bin.
- Carpeta worlds: Contiene nuestro mundo (definición de gravedad, suelo, luz...) con nuestro robot(.world). Dentro de src.
- Carpeta launcher: La carpeta launcher contiene nuestro lanzador, el que instanciamos primero para crear nuestro mundo. Dentro de worlds.
- Archivo package.xml: Proporciona meta información acerca del paquete.
- Archivo CmakeList, que usa catkin para compilar, es una lista en la que se especifica al compilador que debe compilar de nuestro paquete: nodos, mensajes, servicios, librerías, etc.
- Archivo manifest.xml: contiene una descripción del paquete (función, autor, licencia, etc.) y una lista con los paquetes de los que depende. Según la funcionalidad del paquete, es posible encontrarnos más o menos carpetas con archivos diferentes. En el caso más común de un paquete que se usa para simulación en Gazebo.


```

workspace_folder/      -- WORKSPACE
src/                   -- SOURCE SPACE
  CMakeLists.txt       -- The 'toplevel' CMake file
  package_1/
    CMakeLists.txt
    package.xml
    ...
  package_n/
    CMakeLists.txt
    package.xml
    ...
build/                 -- BUILD SPACE
  CATKIN_IGNORE        -- Keeps catkin from walking this directory
devel/                 -- DEVELOPMENT SPACE (set by CATKIN_DEVEL_PREFIX)
  bin/
  etc/
  include/
  lib/
  share/
  .catkin
  env.bash
  setup.bash
  setup.sh
  ...
install/               -- INSTALL SPACE (set by CMAKE_INSTALL_PREFIX)
  bin/
  etc/
  include/
  lib/
  share/
  .catkin
  env.bash
  setup.bash
  setup.sh
  ...

```

Ilustración 19: D.1 Ejemplo paquete Catkin

3. *Manual del programador*

El programador será el encargado de manejar el curso según las funcionalidades de moodle, para acceder a la sesión de administrador deberá usar las claves de la siguiente ilustración.

Nombre de usuario

admin

Contraseña

Cbr.1990

Ilustración 20: Nombre de administrador y contraseña





3.1. Importar máquina virtual

Por desgracia el programa VMware Player carece de la funcionalidad de crear, copiar o exportar máquinas virtuales, pero hemos encontrado una solución que nos va a permitir clonar la máquina virtual creada.

Lo que vamos a realizar literalmente es una copia de los archivos que forman la máquina virtual.

Buscamos la carpeta donde tenemos nuestras máquinas virtuales, en nuestro caso C:\Users\Carmelo\Documents\Virtual Machines y copiamos todos los archivos de la máquina virtual de nuestro proyecto(se llama TFG).

Vemos en la siguiente imagen todos los archivos que contiene.

e equipo > Boot (C:) > Usuarios > Carmelo > Documentos > Virtual Machines > TFG				
Nombre	Fecha de modifica...	Tipo	Tamaño	
564d3293-82a9-aaf8-b211-da48c4d4a1dc...	04/05/2017 13:06	Carpeta de archivos		
TFG.vmdk.lck	04/05/2017 13:06	Carpeta de archivos		
564d3293-82a9-aaf8-b211-da48c4d4a1dc...	04/05/2017 13:06	Archivo VMEM	4.194.304 KB	
autoinst.iso	22/02/2017 12:59	Archivo ISO	40.016 KB	
TFG.nvram	25/04/2017 19:14	Archivo NVRAM	9 KB	
TFG.vmdk	04/05/2017 13:06	VMware virtual dis...	2 KB	
TFG.vmsd	22/02/2017 12:59	Archivo VMSD	0 KB	
TFG.vmx	04/05/2017 13:36	VMware virtual m...	3 KB	
TFG.vmx	09/03/2017 17:08	Archivo VMXF	4 KB	
TFG-s001.vmdk	04/05/2017 13:06	VMware virtual dis...	2.054.592 KB	
TFG-s002.vmdk	04/05/2017 13:06	VMware virtual dis...	2.042.624 KB	
TFG-s003.vmdk	04/05/2017 13:06	VMware virtual dis...	2.048.832 KB	
TFG-s004.vmdk	04/05/2017 13:06	VMware virtual dis...	1.870.464 KB	
TFG-s005.vmdk	04/05/2017 13:06	VMware virtual dis...	1.898.432 KB	
TFG-s006.vmdk	04/05/2017 13:06	VMware virtual dis...	2.059.968 KB	
TFG-s007.vmdk	04/05/2017 13:06	VMware virtual dis...	2.053.760 KB	
TFG-s008.vmdk	04/05/2017 13:06	VMware virtual dis...	2.067.136 KB	
TFG-s009.vmdk	04/05/2017 13:06	VMware virtual dis...	19.136 KB	
TFG-s010.vmdk	04/05/2017 13:06	VMware virtual dis...	320 KB	
TFG-s011.vmdk	04/05/2017 13:06	VMware virtual dis...	64 KB	
TFG-s012.vmdk	04/05/2017 13:06	VMware virtual dis...	320 KB	
TFG-s013.vmdk	04/05/2017 13:06	VMware virtual dis...	320 KB	
TFG-s014.vmdk	04/05/2017 13:06	VMware virtual dis...	320 KB	
TFG-s015.vmdk	04/05/2017 13:06	VMware virtual dis...	320 KB	
TFG-s016.vmdk	04/05/2017 13:06	VMware virtual dis...	320 KB	
TFG-s017.vmdk	04/05/2017 13:06	VMware virtual dis...	320 KB	
TFG-s018.vmdk	04/05/2017 13:06	VMware virtual dis...	320 KB	
TFG-s019.vmdk	04/05/2017 13:06	VMware virtual dis...	320 KB	
TFG-s020.vmdk	04/05/2017 13:06	VMware virtual dis...	320 KB	
TFG-s021.vmdk	04/05/2017 13:06	VMware virtual dis...	320 KB	
TFG-s022.vmdk	04/05/2017 13:06	VMware virtual dis...	64 KB	
vmware.log	04/05/2017 13:06	Documento de tex...	486 KB	
vmware-0.log	25/04/2017 19:14	Documento de tex...	378 KB	
vmware-1.log	25/04/2017 14:06	Documento de tex...	444 KB	
vmware-2.log	25/04/2017 11:36	Documento de tex...	545 KB	
vprintproxy.log	04/05/2017 13:06	Documento de tex...	171 KB	

Ilustración 21: Todos los archivos de nuestra máquina virtual

Tenemos “clonada” la máquina virtual, ahora debemos llevar todos estos archivos a otro equipo donde importarla. En este nuevo equipo arrancamos el VMware Player y clicamos sobre Open a Virtual Machine como en la siguiente imagen.

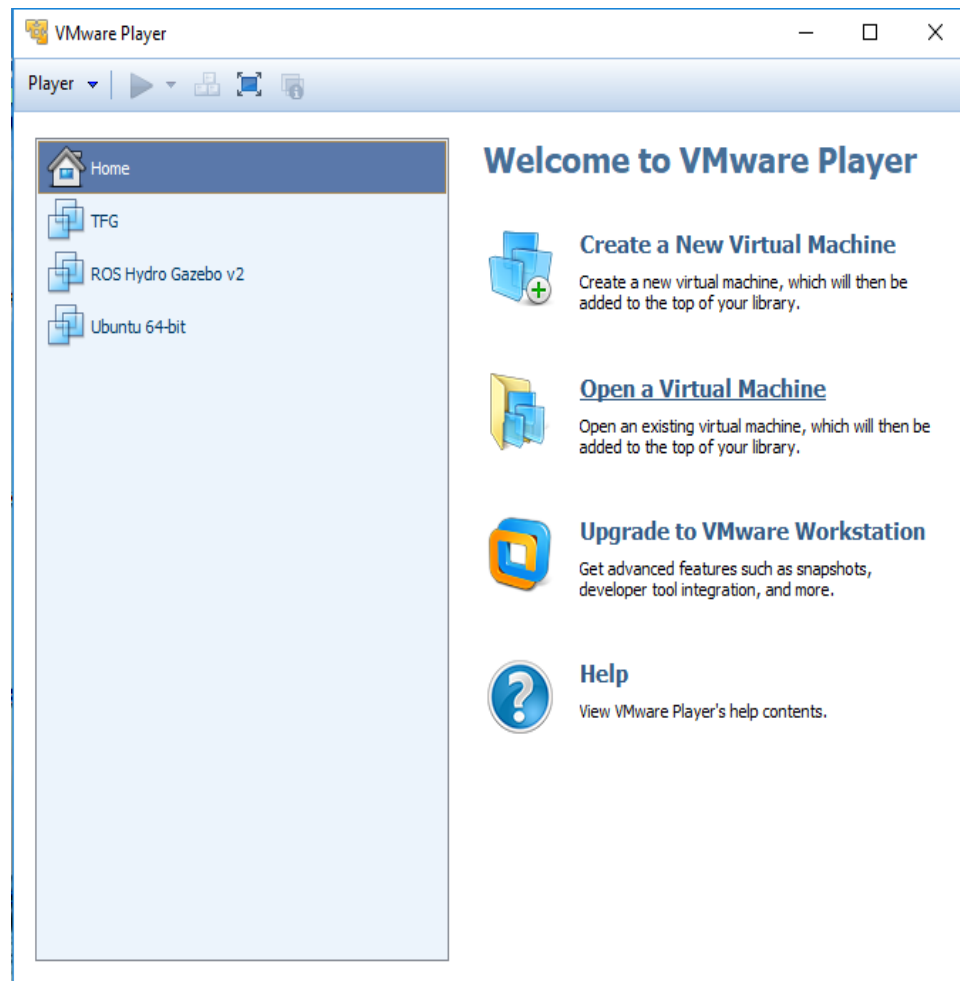


Ilustración 22: VMware Player

Buscamos el fichero de configuración (.vmx) de nuestra máquina virtual, le seleccionamos y pulsamos Abrir.





564d3293-82a9-aaf8-b211-da48c4d4a1dc...	04/05/2017 13:06	Carpeta de archivos	
TFG.vmdk.lck	04/05/2017 13:06	Carpeta de archivos	
TFG.vmx	04/05/2017 13:36	VMware virtual m...	3 KB

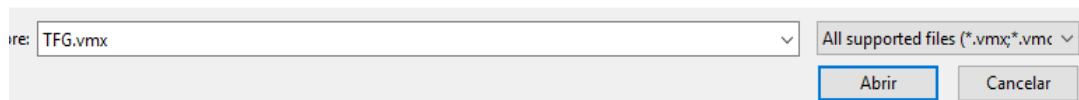


Ilustración 23: Importar máquina virtual

Ya tenemos la maquina virtual copiada en otro equipo, solo queda clicar en play virtual machine, y tendremos nuestra máquina virtual lista.

Para poder usar la sesión con el entorno preparado debemos abrir la sesión que hemos usado con nombre Carmelo, y con contraseña “123456”.

Una vez dentro de nuestra máquina virtual cuya descarga se hará desde mediafire siguiendo el enlace que facilitaremos ahora (también disponible en el foro del curso), la preparación del entorno ya estaría hecha. Para ver como la realizamos ver temas 2 y 3 del curso se explica su realización.

Son 4 partes de 2GB, deberéis descomprimirlo y pegarlo en la carpeta VirtualMachine de vuestro VMWare.

<https://www.mediafire.com/folder/y5xuppheaq81y8q.2dox012bdrxiok8,fyg3yk58xz6ihls,7zh33u22rn7gt0u/shared>

3.2. Instalación de herramientas

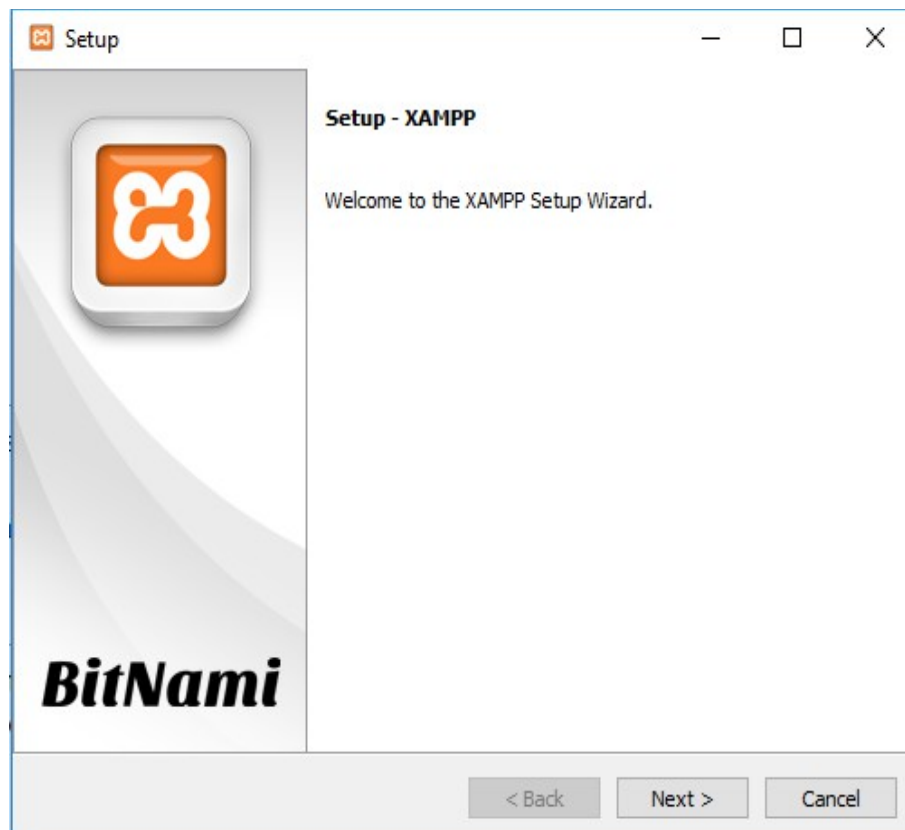
3.2.A. Servidor

XAMPP[3] es una distribución de Apache completamente gratuita y fácil de instalar que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MariaDB, PHP, Perl.

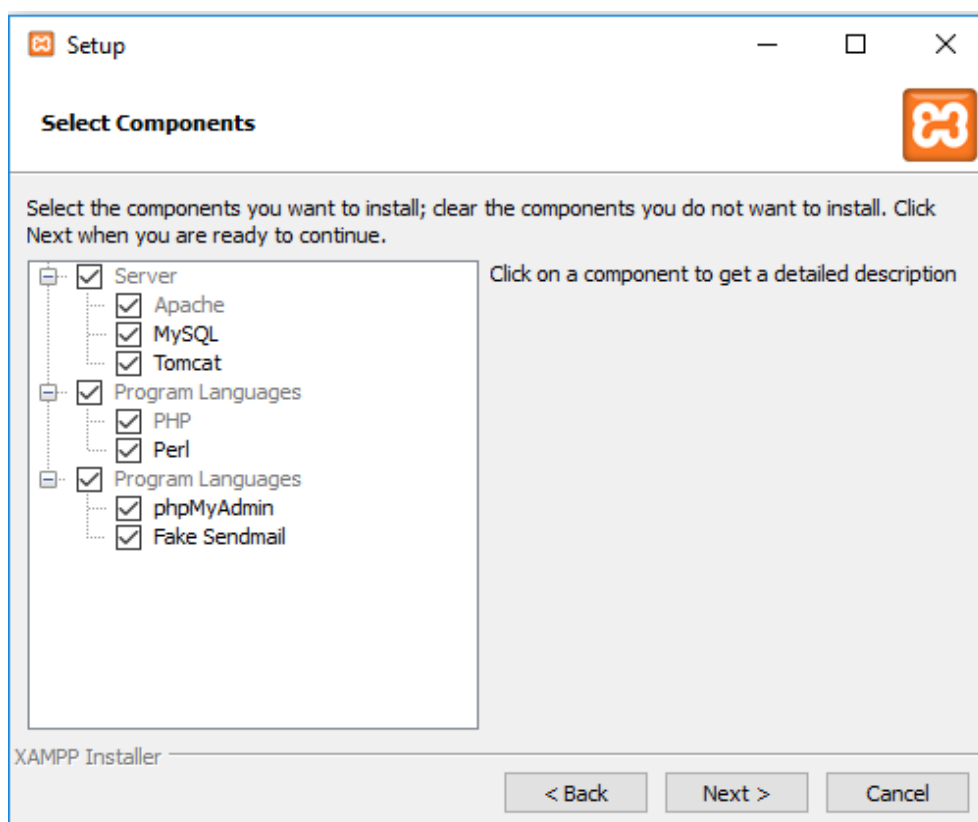
1. Accedemos a la web oficial de Xampp y descargamos el archivo correspondiente a la última versión

https://www.apachefriends.org/es/download_success.html

2. Ejecutamos el fichero descargado .exe:

**Ilustración 24: Instalación xampp**

3. Seleccionamos los componentes que deseamos instalar.

**Ilustración 25: Instalación xmapp componentes**

4. Seleccionamos que nos cree el directorio en C:\xampp

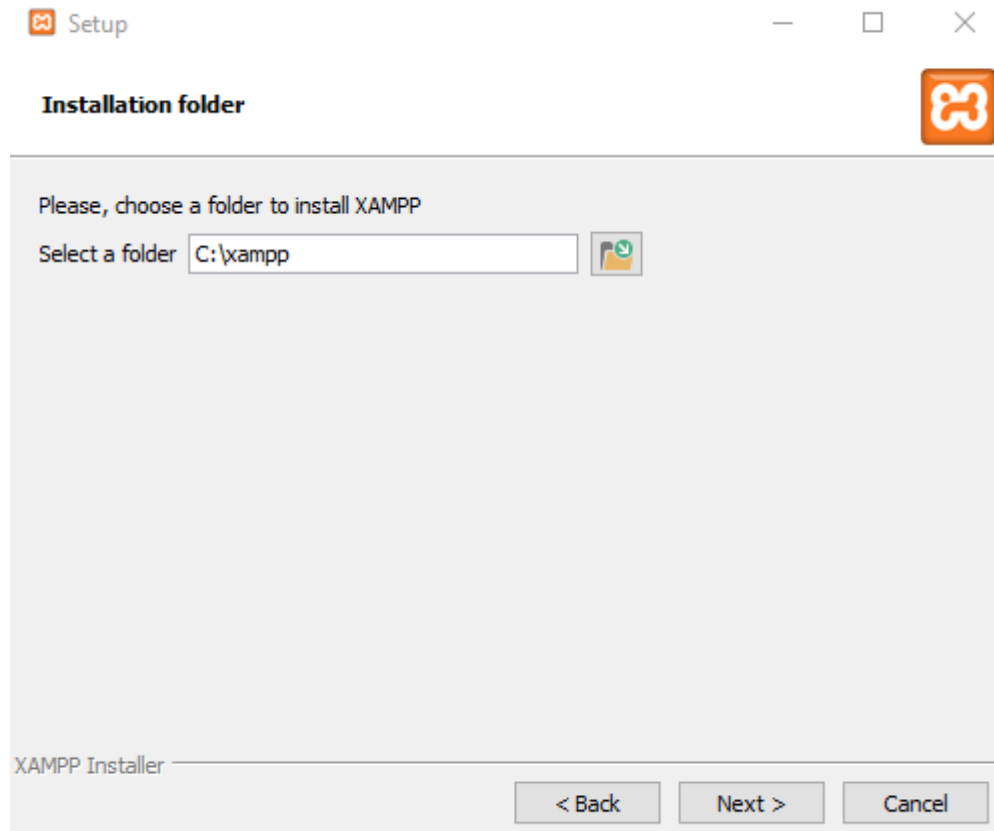


Ilustración 26: Instalación xampp carpeta

5. Se instalarán todos los componentes, y ya tendremos nuestro servidor listo.

3.2.B. Moodle

Vamos a explicar a continuación el proceso de instalación de moodle(versión 3.1.6):

1. Accedemos a la web oficial de Moodle y descargamos el archivo zip correspondiente a la versión 3.1.6:
<https://download.moodle.org/releases/latest/>
2. Copiamos el fichero descomprimido en la siguiente carpeta: “C:\xampp\moodle”
3. Abrimos el navegador e introducimos la siguiente URL para comenzar el proceso de instalación de Moodle: “localhost”
4. Se nos mostrará la siguiente pantalla:



Seleccionar idioma

Por favor, seleccione un idioma para el proceso de instalación. Este idioma se usará también como idioma por defecto del sitio, si bien puede cambiarse más adelante.

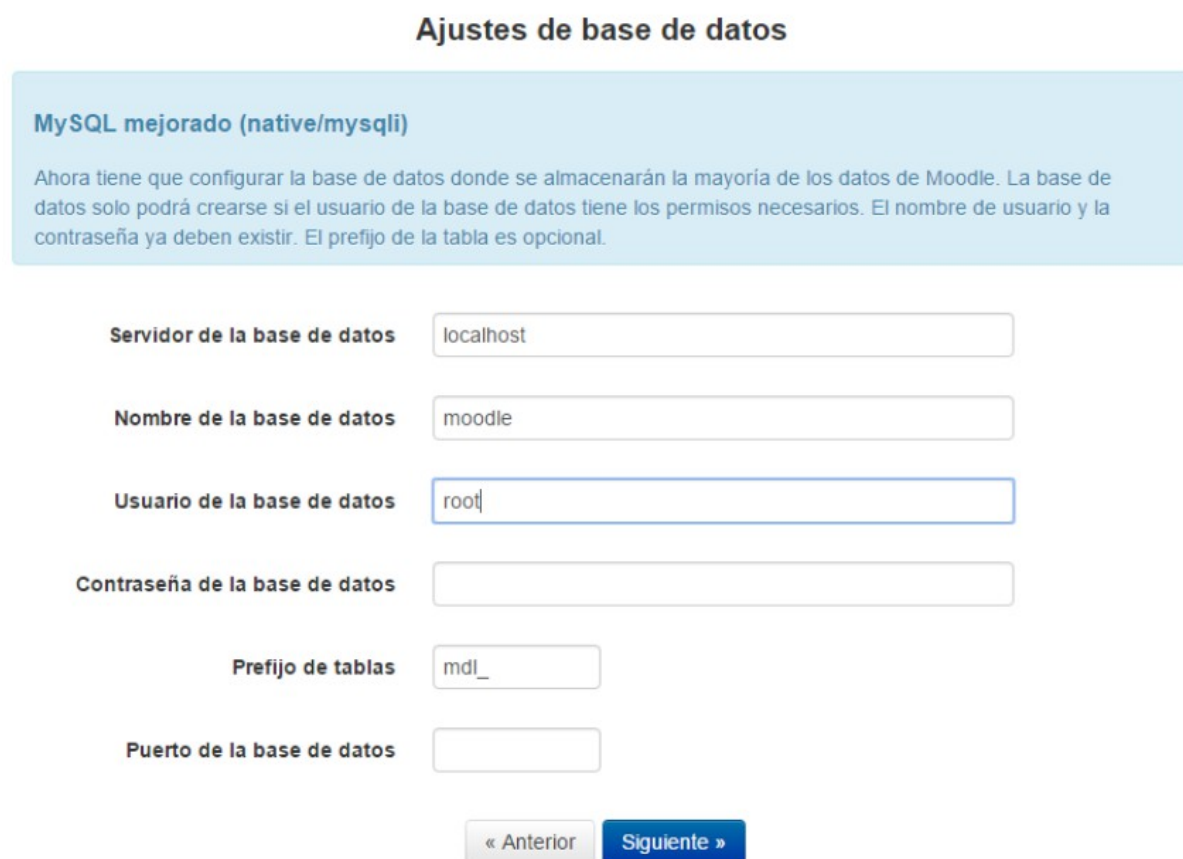
Idioma

Siguiente »



Ilustración 27: Instalación moodle selección idioma

5. Seleccionaremos “Siguiente” dejando todos los valores que vienen por defecto hasta que aparece la siguiente pantalla, en la cual rellenaremos los datos de la siguiente forma:



Ajustes de base de datos

MySQL mejorado (native/mysqli)

Ahora tiene que configurar la base de datos donde se almacenarán la mayoría de los datos de Moodle. La base de datos solo podrá crearse si el usuario de la base de datos tiene los permisos necesarios. El nombre de usuario y la contraseña ya deben existir. El prefijo de la tabla es opcional.

Servidor de la base de datos

Nombre de la base de datos

Usuario de la base de datos

Contraseña de la base de datos

Prefijo de tablas

Puerto de la base de datos

« Anterior **Siguiente »**

Ilustración 28: Instalación moodle ajustes base de datos

6. Pinchamos “Siguiente”, después “Continuar” y el proceso de instalación de Moodle comenzará.
7. Una vez instalado, se nos mostrará la siguiente pantalla donde rellenaremos los campos de la siguiente forma:





Nombre de usuario*

Escoger un método de identificación: ☒ Cuentas manuales ☐ ?

La contraseña debería tener al menos 8 caracter(es), al menos 1 dígito(s), al menos 1 minúscula(s), al menos 1 mayúscula(s), al menos 1 caracter(es) no alfanuméricos

Nueva contraseña* ☐ Desenmascarar

Forzar cambio de contraseña ☐ ?

Nombre*

Apellido(s)*

Ilustración 29: Instalación de moodle datos admin

8. Continuamos el proceso de instalación y por último nos piden los nombre para la página principal, los cuales serán rellenados de la siguiente forma:
 - Nombre completo del sitio: Curso Gazebo ROS
 - Nombre corto del sitio: Gazebo-ROS
9. La instalación de Moodle estaría ya realizada.

3.3. Versiones de fallos

Fallo/Errores	Descripción
Primera instalación en kubuntu	Fue la primera versión para pruebas con nuestro simulador Gazebo, pero era un sistema muy limitado con el que trabajar
Máquina virtual de matlab	Probamos una máquina virtual de la ayuda del programa matlab, pero estaba desactualizada en la versión 6 de gazebo, por lo que pronto la desechamos
Versión ROS	Probamos con la versión Hydro de ROS, pero nos dimos cuenta de que no tenía algunas funciones que aparecían en los manuales de ROS, después probamos con la versión kinetic
Sobrecalentamiento PC	El simulador Gazebo gasta muchos recursos del sistema, y llego a quemarnos la tarjeta de video de la placa base.
Métodos de librería Gazebo	Algunos de las referencias a métodos de las librerías de Gazebo no funcionan correctamente

Tabla 21: Tabla de fallos

Por estos errores ya citados se aconseja seguir el curso creado en las versiones que se aconsejan, dado que sino podríamos encontrarnos con numerosos fallos.

4. Importación y exportación del curso

4.1. Importación

1. Seleccionamos “Restaurar en el menú administración
2. Pulsamos el botón “Subir archivo”
3. Seleccionamos el fichero zip que contiene el curso
4. Pulsamos el botón “Subir archivo”
5. Seleccionamos “Restaurar”
6. Tras un par de confirmaciones indicamos como queremos cargar el curso
7. Pulsamos el botón “Restaurar ahora” y alguna confirmación adicional
8. El curso está importado

4.2. Exportación

1. Conectarnos al curso que queremos exportar
2. En el menú administración seleccionamos “Copia de seguridad”
3. Seleccionamos los elementos que queremos exportar
4. Indicar el nombre de fichero (opcional)
5. Un vez generado el fichero, lo seleccionamos para descargarlo.





V - DOCUMENTACIÓN DE USUARIO

1. *Introducción*

Esta sección se centra en la explicación y demostración de que características debemos tener para poder acceder y realizar el curso.

2. *Requisitos de usuarios*

Esta sección tiene como objetivo indicar aquellos requisitos que necesita el usuario para poder utilizar el proyecto.

2.1. **Requisitos software**

Es necesario tener instalado el gestor de máquinas virtuales VMWare Player. Este gestor puede ser utilizado prácticamente en cualquier sistema operativo, soportando la mayoría de sistemas Windows, Linux, Mac OS X y Solaris. Dado que no podemos clonarla, luego explicaremos bien como la hemos copiado, para más tarde pegarla en otro ordenador.

2.2. **Requisitos hardware**

En la parte hardware seremos un poco exigentes, necesitamos que pueda correr en la máquina virtual sin problemas. Necesitamos mínimo una RAM de 2GB en la máquina virtual, y un disco duro de al menos 20 GB, el procesador del equipo a poder ser que sea potente, en nuestro caso usamos un I7 a 2,2Ghz y debe usarse uno igual o superior.

3. *Manual de usuario*

Nuestro entregable es un curso de moodle por lo que no tendrá un manual de usuario de la aplicación dado que no es de nuestra propiedad, pero para poder ver de manera sencilla el trabajo realizado para el usuario hemos creado el curso en un servidor online gratuito facilitado en el siguiente link: <https://cursogazeboros.milaulas.com/>

Nombre de usuario
alejandrasalmeron

Contraseña
Alumno100%

El formulario muestra dos campos de entrada. El primer campo, etiquetado como 'Nombre de usuario', contiene el texto 'alejandrasalmeron'. El segundo campo, etiquetado como 'Contraseña', contiene el texto 'Alumno100%'. A la derecha del campo de contraseña hay un icono de un ojo, lo que indica la opción de alternar la visibilidad de la contraseña.

Ilustración 30: Nombre y contraseña de un usuario

VI - INFORME

Desarrollaremos un pequeño informe sobre la viabilidad de incorporar la tecnología ROS-Gazebo en una empresa de robótica.

1. Introducción

Los antecedentes de este proyecto se sitúan en el departamento de I+D de la empresa ASTI donde surgió la posibilidad de incorporar simuladores de entorno donde podrían realizar tareas de inspección y mantenimiento de una manera más sencilla y eficiente.

La introducción de un simulador en la empresa podría facilitar en gran medida todas las tareas que se desarrollan en la empresa, ya sea desde el punto de las pruebas de programación, así como prevenir errores, inspeccionar máquinas...

El simulador propuesto es Gazebo, que destaca por tener un motor de renderizado avanzado, soporte para plugins, programación en la nube, un enorme repositorio con la mayoría de robots comerciales y una extensa gama de sensores y cámaras.

Como complemento ideal de este simulador, aparece el sistema operativo para robots ROS el cual está creciendo a un ritmo trepidante. Cada vez más y más investigadores de todo el mundo se están sirviendo de las ventajas que ROS ofrece para sus propios proyectos. De hecho, según algunas fuentes, solo en 2016 se han destinado más de 150 millones de dólares como capital de riesgo en negocios fundamentados en ROS, lo cual dice mucho sobre la relevancia de este sistema operativo.

La viabilidad de llevar este avance tecnológico a cabo depende de las dificultades que pueda haber en el desarrollo de esta tecnología, las dificultades de adaptarlo a una empresa, así como de lo costoso que es aprender un sistema tan innovador y del que no se tienen muchas referencias.

En este informe con lo aprendido durante estos cuatro meses, intentaremos ver los puntos más relevantes, para posteriormente evaluar su viabilidad para apostar por ello de manera futura.



Ilustración 31: Visitantes de ROS por países





1.1. Objetivos

El objetivo principal de este informe es valorar la viabilidad de incorporar el sistema operativo ROS y un simulador de entornos como podría ser Gazebo en una empresa de logística interna. Los puntos más destacables de incorporar estas tecnologías son:

- Trabajar sin necesidad de tener un entorno similar al del cliente: Podría darse el caso de que un robot para un cliente tuviese que actuar sobre unas condiciones físicas extremas, y quizás no tengamos medios suficientes para poder poner dichas condiciones, mientras que en un simulador sería más sencillo.
- Evitar posibles daños: Los robots podrían tener daños por una mala programación o una inclemencia del entorno
- Trabajar sin necesidad de tener el robot: No necesitamos contar con el robot para poder trabajar con el, solo debemos tenerlo diseñador en el simulador y trabajará
- Mejorar el trato con el cliente: Podríamos simular al momento el entorno en función de las necesidades del cliente, de tal manera que se podrían ir mandando versiones del simulador sobre como se va avanzando.

Por lo cual podríamos decir que el objetivo es que en un periodo de tiempo de 6 meses, los programadores hayan diseñado todos los robots existentes en la empresa, así como incorporado los sensores necesarios, y se pueda empezar a hacer pruebas validas sobre los robots.

1.2. Metodología

Inicialmente la metodología de implantación para todos los programadores sería seguir el curso que hemos realizado en la plataforma Moodle, para así contar con unos conocimientos básicos sobre el entorno a utilizar, y su funcionamiento.

Más tarde sería conveniente crear un grupo de trabajo y así poder ir avanzando colectivamente en los diseños, ir realizando pruebas, para más adelante ir incorporando robots en el simulador.

Finalmente se deberá conseguir un lenguaje de comunicación con el robot por medio de ROS y comprobar que la programación del robot real y el simulado es similar ante la misma programación.

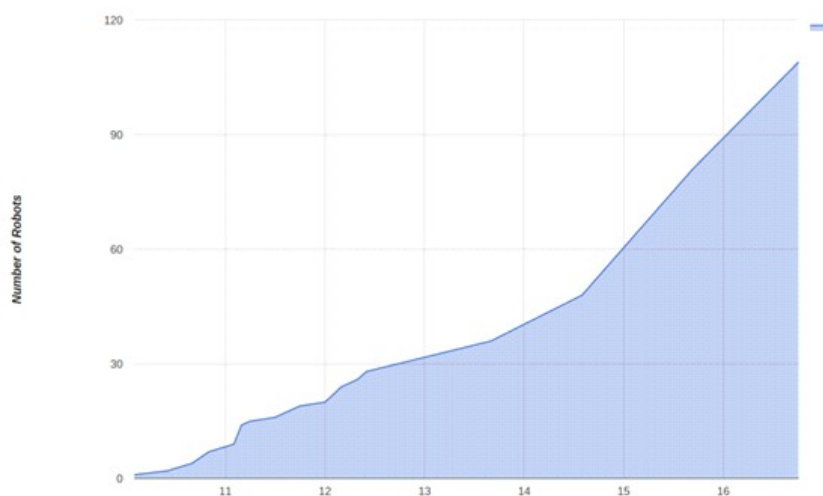


Ilustración 32: Número de robots creados por año

1.3. Contenidos

En la implantación de estos sistemas, enumeraremos los puntos a tener en cuenta:

- Preparación del entorno
- Simulación de robots
- Creación de robots
- Comunicación con robot(ROS)
- Pruebas con robots creados

Para todo ello usaremos un sistema operativo ubuntu, un simulador Gazebo, un sistema operativo para el robot (ROS), y un visualizador de fuerzas RVIZ. Estas serían las herramientas principales a usar.

1.4. Temporalidad

Según diferentes artículos y estudios que hemos podido ver, se presupone que el tiempo de aprendizaje para ROS únicamente sería de 12 meses. En cambio teniendo en cuenta que el principio es lo más complejo, contando con que tenemos un buen curso con el que acortar los plazos, y que se trabajará en un grupo por lo que se avanzará colectivamente, podríamos estimar que serán 6 meses de aprendizaje.

En segundo lugar estaría el tiempo necesario de implantación, que dependerá de si se busca una implantación total, o parcial (en todos los robots y departamentos de la empresa o solo en algunos).

2. Estudio Mercado

2.1. Empresas del sector

2.1.A. JBT

Es una empresa de soluciones tecnológicas diseñan diferentes tipos de AGVs, para empresas, hospitales, farmacéuticas...

Según se puede apreciar en su página web trabajan con simuladores similares a Gazebo, como vemos en la siguiente ilustración, por lo que podríamos decir que un simulador sería ventajoso para este tipo de empresas.

Podemos saber más sobre esta empresa en su página web: www.jbtc-agv.com





Ilustración 33: Simulador de JBT

2.1.B. Egemin

Dematic Egemin es un proveedor de vanguardia de los sistemas estándar de vehículos automatizados guiados (sistemas AGV) para aplicaciones específicas.

Cuentan con un software propio para la navegabilidad y la gestión de los AGV, por lo que no tenemos información de si está basado en ROS o no.

Podemos saber más sobre esta empresa en su página web:

<http://www.egemin-automation.com/en/>

2.1.C. Electric-80

Es una empresa enfocada en producir AGVs para las empresas de bienes de consumo de movimiento rápido, especialmente en los sectores de alimentos, bebidas y tejidos. Buscan mejorar constantemente sus conocimientos tecnológicos, de ingeniería e informática para optimizar la gestión de todos los procesos.

Tanto es así que según su página web podemos ver que ofrecen al cliente servicios de simulación para diseñar su entorno, como podemos ver en la siguiente imagen.

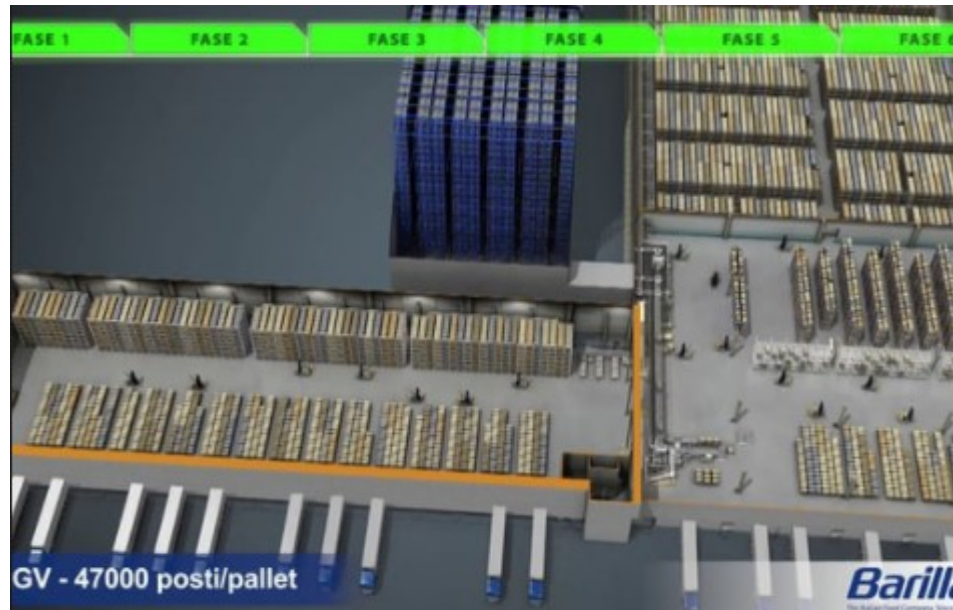


Ilustración 34: Simulador de Electric-80

Podemos saber más sobre esta empresa en su página web:

<http://www.electric80.com/it/>

2.2. Otros

Hemos encontrado otras empresas que usan la tecnología que deseamos implantar para sus negocios, vamos a enumerarlas:

- erle robotics: Ejemplo de una empresa que se dedica al mundo de la programación de drones y trabaja con ROS.



Ilustración 35: Erle robotics





- Diferentes start-ups focalizadas en robótica móvil, se encargan de poner en contacto a empresarios y programadores para trabajar, en algunas soluciones que han desarrollado se puede ver el uso de ROS. <http://www.startupedmonton.com/>
- Diferentes grupos de investigación que desarrollan sobre ROS.

3. Recursos humanos

La idea sería contar con un grupo de trabajo de investigación para poder ir avanzando de manera conjunta en el mundo ROS, siendo importante la cooperación entre los integrantes de este grupo para el aprendizaje en estas tecnologías.

El grupo podría contar con 5 personas que se adjudicarían principalmente de avanzar en el mundo del modelado con Gazebo, más adelante se encargarían nuevas funciones a este grupo.



Ilustración 36: Puntos en los que se crean robots

3.1. Conocimientos previos necesarios

En el grupo de trabajo, todos deberían ser titulados en grado de ingeniería informática, o en su defecto podrían ser ingenieros con un máster en algún ámbito relacionado con la programación. Pero se necesitaría un alto grado de nivel como programador.

4. Plan financiero

4.1. Costes económicos de licencias y hardware

Herramienta	Programa	Coste
Sistema Operativo anfitrión	Linux	0€
Sistema operativo robótico	ROS	0€
Simulador	Gazebo	0€
IDE	Eclipse	0€
Ordenador	Ordenador de trabajadores	5*750€= 3750€
TOTAL:		3750€

4.2. Costes económicos de trabajadores

Según las estimaciones que hemos podido hacer, creemos que en 6 meses el equipo de trabajo ya tendría un nivel alto en el manejo de estos programas, además hemos estimado que el sueldo de cada persona sea de 17€/hora, y decidimos con anterioridad que el grupo de trabajo sería de 5 personas, por lo que:

6 meses son unas 18 semanas a 40 horas semanales son unas 720 horas.

720 horas a 17€ la hora son 12240€.

Serían 5 los trabajadores destinados al grupo de trabajo, por lo que 12240€ por 5 trabajadores serían 61200€

En seguridad social habría que pagar aproximadamente un 30% de lo que ingresan adicionalmente, por lo que finalmente serían 79560€

4.3. Previsión de ingresos y amortización

Teniendo en cuenta un artículo encontrado en Internet que dice que el AGV vale a partir de 12000€ lo que tú quieras gastar [4] vamos a suponer que un coste medio de un AGV será 20000€, por lo cual sería amortizado a partir del cuarto AGV que se venda.

Es difícil estimar cuánto tiempo se gana con la programación en un simulador pero estimaremos que los robots se programan ahora más rápido, y que se producen menos errores en la programación que ocasionen revisiones. Es por ello que podremos ahorrar unos 1500€ de cada robot entre fallos y tiempo perdido, así que para amortizar los 83310€ (3750€ + 79560€) que nos costaría implantar la tecnología necesitaríamos vender 55AGVs.

4.4. Previsión de nuevos gastos

En cuanto a los cálculos que hemos hecho de implantación no hemos tenido en cuenta que los ordenadores asignados al equipo de desarrollo tienen una vida útil de 4 años, por lo que podremos decir que los nuevos gastos serán en un periodo aproximado de 5 años de:

5 equipos * 750€ por equipo 3750€

Es decir, cada 5 años se deberá gastar unos 3750€ en nuevos equipos para los programadores.

5. Conclusión

Después de realizar este pequeño informe abordando los puntos más importantes a tener en cuenta en la implantación de estas tecnologías en una empresa, podemos finalizar afirmando que es altamente ventajoso el uso de una simulación en los comportamientos de los AGVs.

Según hemos podido ver, otras empresas del sector ya usan herramientas similares para la simulación de sus entornos, de tal manera que sería ventajoso para cualquier empresa del sector implantar la tecnología.

Como hemos podido observar en diferentes gráficas, el sistema ROS está siendo usado cada día por más personas y empresas.

En cuanto al precio de implantación podríamos decir que no es un precio, sino una inversión a medio-largo plazo, puesto que solo tendríamos que realizar una inversión inicial para pagar al grupo de trabajo durante 6 meses (donde no serán productivos), pero luego los gastos





serían insignificantes.

Por último, desde un punto de vista personal y después de haber estado trabajando estos programas, viendo su buen funcionamiento, y las ventajas que podría dar a una empresa de este tipo aconsejo basándome en este informe su implantación.

VII - REFERENCIAS

Bibliografía

- [1] Gurley, Gabriel: «A Conceptual Guide to OpenOffice.org 2 for Windows and Linux» 2007
- [2] colaboradores de Moodle, Diseño curso, 19 feb 2017 a las 14:21 UTC, https://docs.moodle.org/all/es/Arquitectura_de_Moodle#Debe_ser_f.C3.A1cil_de_actualizar_desde_una_versi.-C3.B3n_a_la_siguiente
- [3] colaboradores de Wikipedia, Xampp, 2 de julio del 2017, 22:18 UTC, <https://es.wikipedia.org/wiki/XAMPP>
- [4] Soraya Paniagua, ASTI, la pyme española líder europea en Vehículos de Guiado Automático, 3 de julio del 2017, 06:13 UTC, <http://www.sorayapaniagua.com/2016/02/15/asti-la-pyme-espanola-lider-europea-en-vehiculos-de-guiado-automatico/>





Impreso en Burgos el lunes, 3 de julio de 2017